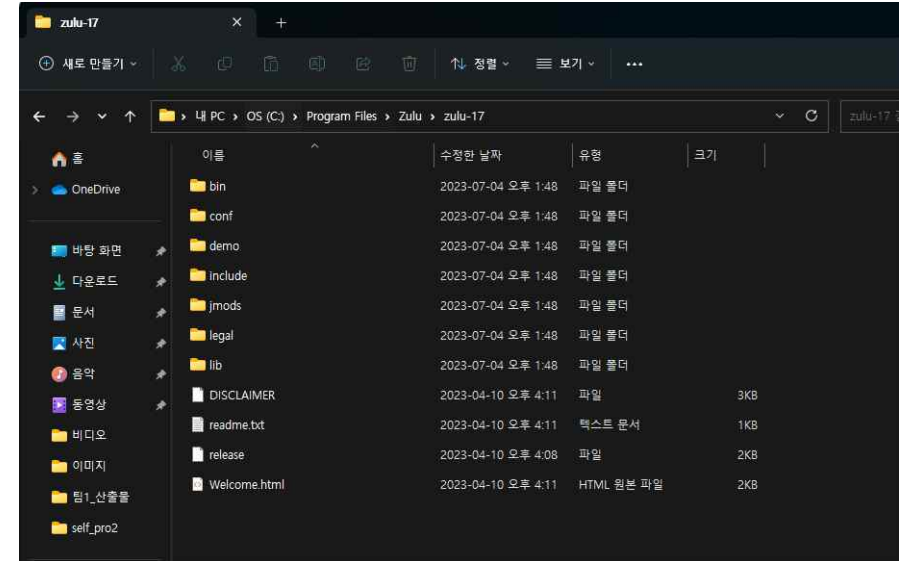
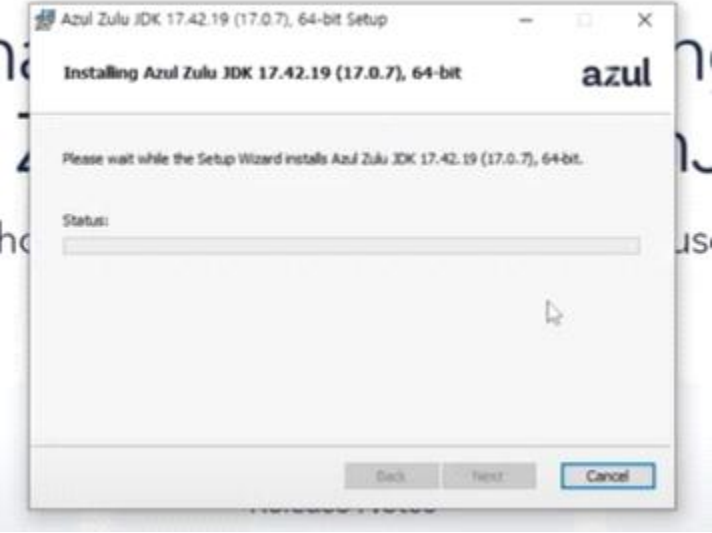
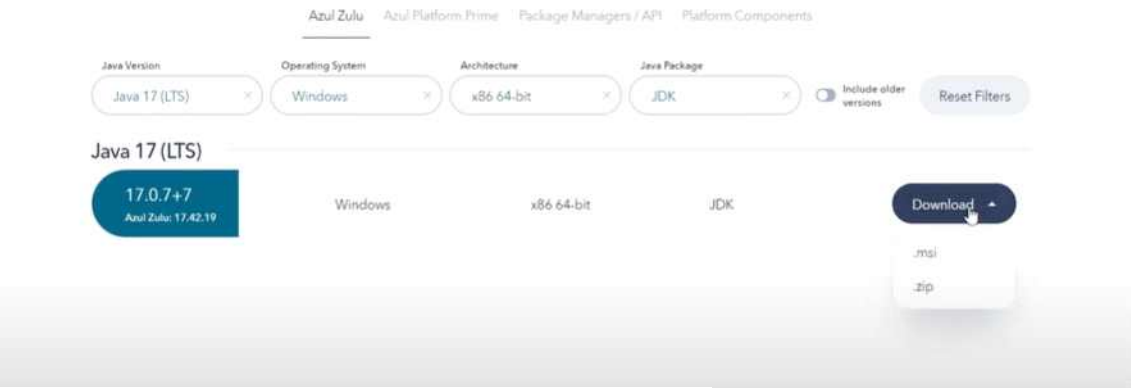


과정명	에듀테크 풀스택 자바 개발자 양성 과정			평가일시	2023년 09월 04일
과목명	웹 개발 프로그래밍	능력단위요소	개발환경 구축하기	작성자명	오세훈
			공통 요소 구현하기		
			처리 기능 구현하기		
			화면 구현하기		
			배치 프로그램 구현하기		

해당 능력단위 요소	개발환경 구축하기	(20점)
수행1. 작성 내용		

1. 자바 개발환경 구축



2. ide

mvn

사용한 라이브러리의 mvn 연결 주소

com.google.code.gson:gson:2.7

javax.json:javax.json-api:1.1.4

javax.servlet:javax.servlet-api:4.0.1

org.json:json:20151123

org.jsoup:jsoup:1.12.1

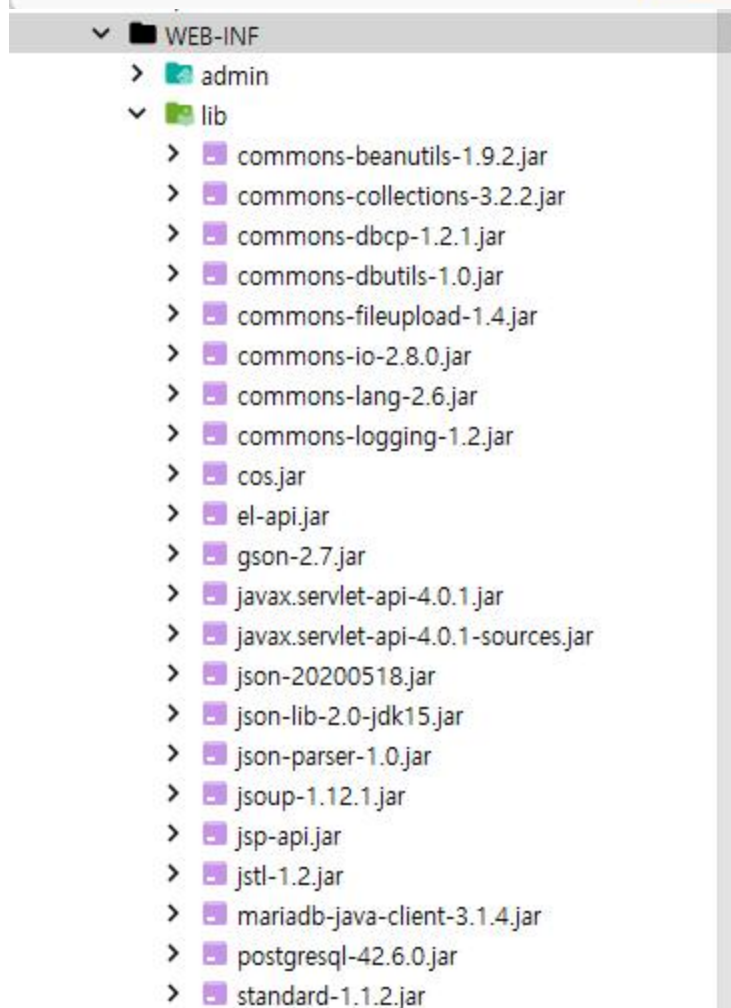
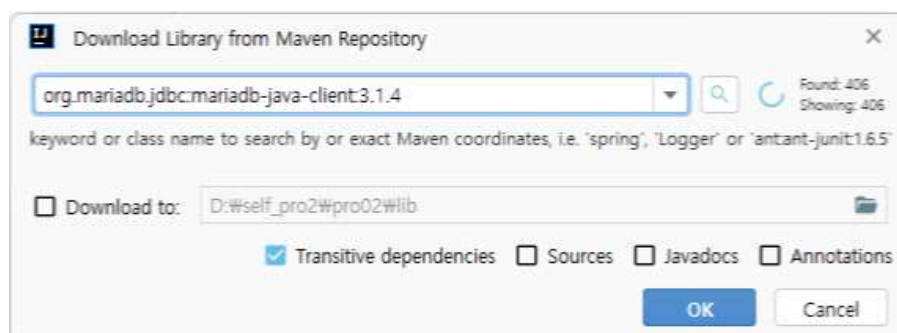
jstl:jstl:1.2

junit:junit:4.13

org.mariadb.jdbc:mariadb-java-client:3.1.4

net.markenwerk:utils-json-parser:1.0.0

org.postgresql:postgresql:42.6.0



3. 데이터 베이스 설치

This is a short term release which is maintained for one year only. The latest long-term release is MariaDB 10.11, which is maintained for five years.

MariaDB Server Version

MariaDB Server 11.0.2

Display older releases: ☐

Operating System

Windows

Architecture

x86_64

Package Type

MSI Package

Download

Mirror

Blendbyte - Taipei

Release date: 2023-06-07
File name: mariadb-11.0.2-winx64.msi
File size: 62.1 MB
• Download galera-26.4.14
Display signature and checksums

Database settings

Default instance properties

MariaDB 10.11 (x64) database configuration

MariaDB Server

☒ Install as service

Service Name: MariaDB

☒ Enable networking

TCP port: 3306

InnoDB engine settings

Buffer pool size: 2039 MB

Page size: 16 KB

Back

Next

Cancel

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.007 sec)

MariaDB [(none)]> |
```

```
MariaDB [(none)]> CREATE DATABASE test1
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right syntax
to use near 'DATE' at line 1
MariaDB [(none)]> create database test1;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test1 |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> |
```

4. 자바 웹 개발 환경에서 db 연동 테스트

```
public class MariaDBCon implements DBConnect {

    1 usage
    ⚡ final String DRIVER = "org.mariadb.jdbc.Driver";    Sedoll, 2023
    1 usage
    final String URL = "jdbc:mariadb://localhost:3306/self2";
    1 usage
    final String USER = "root";
    1 usage
    final String PASS = "1234";
    2 usages
    Connection conn = null;

    Custom{id='kim', pw='PoFwcUNmztMSrVIZNnSjPgluwbPVYHUiSzicdxofwckMOUmrtQZNWN0Iv1kyht5PvqAAUg==', name='김철수',
    Custom{id='hong', pw='PoFwcUNmztMSrVIZNnSjPgluwbPVYHUiSzicdxofwckMOUmrtQZNWN0Iv1kyht5PvqAAUg==', name='홍길동',
```

Show entries

번호	아이디	이름	가입일	비고
1	kim	김철수	2023-09-01 16:00:14	<button>탈퇴</button>
2	hong	홍길동	2023-09-01 16:00:14	<button>탈퇴</button>

Search:

Previous 1 Next


```

<link rel="stylesheet"href="${path}/css/font.css">
<!-- css 모듈화 -->
<link rel="stylesheet"href="${path}/css/common.css">
<link rel="stylesheet"href="${path}/css/hd.css">
<link rel="stylesheet"href="${path}/css/ft.css">
<link rel="stylesheet"href="${path}/css/change_img.css">
<link rel="stylesheet"href="${path}/css/slidebox.css">
<style>
        .img_box      li.item1      .bg_boxbackground-image:
url("${path}/img/006.png");}
        .img_box      li.item2      .bg_boxbackground-image:
url("${path}/img/008.png");}
        .page{clear: both; width: 100vw; height: 100vh; position: relative;}
        .page::after{content: ""; display: block; width: 100vw; clear: both;}
        .page_wrapclear: both; width: 1200px; height: auto; margin: 0 auto;}
        .page_tit{font-size: 48px; text-align: center; padding-top: 50px;
padding-top: 0.5em;}
        #page2{height: 400px; background-color: #f5f9fe; background-position:
15vw 75vh;
        background-repeat: no-repeat; padding-top: 100px;}
        .pic_lst{clear: both; width: 1200px; margin: 150px auto;}
        .pic_lst li{width: 280px; height: 400px; margin-left: 26px; float:
left; background-repeat: no-repeat;
        background-position: center center; background-size: cover;
filter: blur(1px) brightness(80%); box-shadow: 12px 12px 12px #999;}
        .pic_lst li adisplay: block; width: 256px; height: 376px; margin:
11px;}
        .pic_lst li a:hover{border: 1px solid #fff; transition: 0.3s;}
        .pic_lst li:first-child{margin-left: 0;} /* 안쪽에만 마진이 있으므로 테
두리 부분의 마진 제거*/
pic_lst li.item1background-image: url("${path}/img/img_social_main01.jpg");}
        .pic_lst      li.item2background-image:
url("${path}/img/img_social_main02.jpg");}
        .pic_lst      li.item3background-image:
url("${path}/img/img_social_main03.jpg");}
        .pic_lst      li.item4background-image:
url("${path}/img/img_social_main04.jpg");}
        .pic_lst li:hovermargin-top: -20px; transition: 0.5s; filter:
brightness(105%); } /*호버 후 움직여서 밝기*/
        .pic_com{padding-left: 20px; padding-top: 20px; font-size: 14px;}
        .pic_tit{padding-left: 20px; padding-top: 20px; font-size: 18px;}
        .pic_arrow{display: block; width: 110px; height: 10px; border-bottom:
1px solid white; margin-left: -20px; margin-top: 20px;
        transition: 0.4s; position: relative;}
        .pic_lst li:hover .pic_arrow{margin-left: 20px;}
        .pic_arrow::after{content: ""; width: 10px; height: 10px; position:
absolute; right: 0; top: 0; border-bottom: 1px solid white;
        display: block; clear: both; transform-origin: 100%100%;
transform: rotate(45deg); display: none;} /* 화살표 생성 */
        .pic_lst li:hover .pic_arrow::after{display: block;} /*처음엔 안보였다

```


가 호버하면 화살표 생성 */

```
                .card_lst      li.item2      .thumb_box{background-image:
url("${path}/img/t1.jpg");}
                .card_lst      li.item3      .thumb_box{background-image:
url("${path}/img/t2.jpg");}
                .card_lst      li.item4      .thumb_box{background-image:
url("${path}/img/t3.jpg");}
                .card_lst      li.item5      .thumb_box{background-image:
url("${path}/img/t4.jpg");}
                .card_lst      li.item6      .thumb_box{background-image:
url("${path}/img/t5.jpg");}
                .card_lst      li.item7      .thumb_box{background-image:
url("${path}/img/t6.jpg");}
```

```
.ico.item1{background-position: -70px -60px;} /*네이버 블로그 아이콘*/
.ico.item2{background-position: -140px -120px;} /*인스타 그램 아이콘*/
```

/*중간*/

```
.board_tit{font-size: 38px; text-align: center;
padding-top: 50px;
padding-top: 0.5em;
margin-bottom: 20px}
.latest_board{
width:1000px;
margin: 0 auto;
}

.leftcontent{
width: 500px;
height: 700px;
float: left;
}

.rightcontent{
width: 500px;
height: 700px;
display: inline-block;
}

.rightcontent table{
float: right;
}

.tb1{
font-size: 16px;
width: 480px;
}

.tb1 th{
```



```

        font-weight: bold;
        background-color: #dae4f3;
        line-height: 40px;
        border-right: 1px solid white;
    }

    .tb1 td{
        line-height: 34px;
        border-bottom: 1px solid #d2d2d2;
    }
</style>
</head>
<body>
<header class="hd" id="hd">
<%@include file="/header.jsp"%>
</header>
<div class="contents">
    <figure class="vs">
        <ul class="img_box">
            <li class="item1 active">
                <input
type="radio" name="vs_ra" id="vs_ra1" class="vs_ra" checked>
                <div class="bg_box"></div>
<!--
                <h2 class="vs_tit">“나눔의 힘! 실천의 힘!”<br>--%>
<!--
                <strong>작은 따뜻함으로 세상을 <br>바꿉니다</strong>--%>
<!--
                </h2>--%>
            </li>
            <li class="item2">
                <input type="radio" name="vs_ra" id="vs_ra2" class="vs_ra">
                <div class="bg_box"></div>
<!--
                <h2 class="vs_tit">교육에 대한 끊임없는 도전<br>--%>
<!--
                <strong>행복한 내일을 함께 합니다</strong></h2>--%>
            </li>
        </ul>
        <ul class="btn_box">
            <li class="item1 active"><label
for="vs_btn1" class="vs_btn"></label></li>
            <li class="item2"><label
for="vs_btn2" class="vs_btn"></label></li>
        </ul>
        <button type="button" class="play_btn"></button>
    </figure>
    <script src="./js/rotation.js"></script>
</div>
<footer class="ft" id="ft">
<%@include file="/footer.jsp"%>
</footer>
<a href="#" class="totop">위</a>
</body>
</html>

```

head.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<!-- 나중에 jsp가 헛갈릴 수 있으므로 path를 여러개 사용하는 경우 이름을 다르게 지정
해준다.
path를 지정할 때 이름을 같게 하지 않게해라 같게하면 오류 생길 수도 있음
--%>
<!-- 이미지 경로 바꾸면 오류 생기므로 일단 이렇게 두자 --%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path2" value="<%=request.getContextPath() %>" />
<%
String path1 = request.getContextPath();
String sid = (String) session.getAttribute("id");
String sname = (String) session.getAttribute("name");
String sjob = (String) session.getAttribute("job");
%>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-compatible" content="IE=edge"><!-- 익스프로어 일 경우 옛
지를 사용하라고 안내-->
<meta http-equiv="author" content="TeenTalk"><!-- 문서의 제작사 -->
<meta http-equiv="publisher" content="TeenTalk"><!-- 문서의 저자 정의 -->
<meta http-equiv="copyright" content="copyright@teentalk.co.kr"><!-- 저작권 명
시 -->
<title>TEENBOOK</title>
<!-- 검색엔진 최적화(SEO) -->
<meta name="subject" content="TeenTalk"><!-- 페이지 주제 지정 http-equiv 로 지정
해도 상관 무 -->
<meta name="keywords" content="TeenTalk"><!-- 검색 단어 지정 -->
<!-- 검색 결과에 표시되는 문자 지정-->
<meta name="description" content="TeenTalk은 IT 교육에 대한 자료 제공과 교육을 실
시하는 회사입니다.">
<meta name="robots" content="index, follow">
<!-- 오픈 그래프 -->
<!-- 페이지를 공유할 때 미리보기로 보여주는 코드 -->
<meta property="og:type" content="website">
<meta property="og:title" content="TeenTalk">
<meta property="og:description" content="앞선 생각으로 더 큰 미래의 교육을 준비하
는 기업">
<meta property="og:image" content="<%=path1%>/img/img1.jpg">
<meta property="og:url" content="<%=path1%>/img/img1.jpg">
<!-- 파비콘 -->
<link rel="shortcut icon" href="./img/logo.ico">
<!-- 애플 계열 모바일 -->
<!-- <link rel="apple-touch-icon-precomposed" href="../img/logo_64.png"> -->
<!-- IE 계열 브라우저 -->
<meta name="msapplication-TileColor" content="#FFFFFF">
<meta name="msapplication-TileImage" content="./img/logo_48.png">
<!-- 파이어폭스, 오페라, 또는 구형 크롬/사파리 -->
<link rel="icon" href="./img/logo_16.png" sizes="16x16">
```

```

<link rel="icon"href="./img/logo_32.png"sizes="32x32">
<link rel="icon"href="./img/logo_48.png"sizes="48x48">
<link rel="icon"href="./img/logo_64.png"sizes="64x64">
<!-- 제이쿼리 -->
<script src="./js/jquery-1.10.0.js"></script>
<!-- 부트스트랩 -->
<!--
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"rel="stylesheet"integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHekRQN+PtmoHDEXuppvnDJzQIu9"crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"crossorigin="anonymous"></script>
--%>
<!-- 불법 복제 방지, 필수 -->
<script>
    // document.oncontextmenu = function () {return false;}
    // document.ondragstart = function () {return false;} // 드래그 방지
    // document.onselectstart = function () {return false;}
    // document.onkeydown = function () {return false;} // 키보드 입력 방지
</script>

```

header.jsp

```

<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path2"value="<%=request.getContextPath() %>"/>
<div class="hd_wrap">
    <nav class="tnb"><!-- .hd_wrap:first-child -->
        <ul class="menu">
            <c:if test="${!empty sid}">
                <li>안녕하세요, ${sid}님</li><!-- 로그인 한 회원의 이름 -->
                <li><a href="${path2}/Logout.do">로그아웃</a></li>
            </c:if>
            <c:if test="${empty sid}">
                <li><a href="${path2}/Login.do">로그인</a></li>
                <li><a href="${path2}/Join.do">회원가입</a></li>
            </c:if>
        </ul>
    </nav>
</div>
<div class="hd_wrap"><!-- .hd_wrap:first-child -->
    <a href="/pro02"class="logo">
        
    </a>
    <nav class="gnb">
        <ul class="menu">

```

```

        <li class="item1">
            <a href="{path2}/ProList.do" class="dp1">전체 도서</a>
            <ul class="sub">
                <li><a class="move" href="{path2}/ProList.do?cate=초등">초
등 도서</a></li>
                <li><a class="move" href="{path2}/ProList.do?cate=중등">중
등 도서</a></li>
                <li><a class="move" href="{path2}/ProList.do?cate=고등">고
등 도서</a></li>
            </ul>
        </li>
    <%--
        <li class="item3">
            <a href="{path2}/ProList.do?cate=중등" class="dp1">중등</a>
            <ul class="sub">
                <li><a class="move" href="/qna_problem/qnaList.jsp">문제
QnA</a></li>
                <li><a class="move" href="/qna_career/qnaList.jsp">진로 상
담</a></li>
            </ul>
        </li>
        <li class="item4">
            <a href="{path2}/ProList.do?cate=고등" class="dp1">고등</a>
            <ul class="sub">
                <li><a class="move" href="{path2}/NoticeList.do">공지사항
</a></li>
                <li><a
class="move" href="{path2}/QnaList.do">QnA</a></li>
            </ul>
        </li>--%>
        <li class="item2">
            <a href="" class="dp1">고객 센터</a>
            <ul class="sub">
                <li><a class="move" href="{path2}/NoticeList.do">공지사항
</a></li>
                <li><a
class="move" href="{path2}/QnaList.do">QnA</a></li>
            </ul>
        </li>
        <c:if test="{not empty sid}">
            <li class="item3">
                <a href="{path2}/Mypage.do" class="dp1">마이 페이지</a>
                <ul class="sub">
                    <li><a href="{path2}/CartList.do">장바구니</a></li>
                    <li><a href="{path2}/PaymentList.do">결제내역
</a></li>
                </ul>
            </li>
        </c:if>
        <c:if test="{sid eq 'admin'}">

```

```

        <li class="item4">
            <a href="/"class="dp1">관리자 페이지</a>
            <ul class="sub">
                <li><a href="${path2}/Manage.do">회원탈퇴</a></li>
                <li><a href="${path2}/Sales.do">판매량</a></li>
                <li><a href="${path2}/AdminDeliveryList.do?pstate=0">
배송관리</a></li>
            </ul>
        </li>
    </c:if>
</ul>
</nav>
<script src="/js/load.js"></script>
</div>

```

footer.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<div class="footer_wrap" id="footer">
    <div class="wrapper">
        <div class="footer_logo">
            
        </div>
        <div class="txt_bx">
            <div class="add_bx">
                <span>
서울시 금천구 가산로9길 54 TeenTalk
                </span>
                <br class="show_480">
                <span class="no_be_480">
                    <a href="tel:07041753333">
TEL : 070.7777.3333
                </a>
                </span>
                <span>
FAX : 02.9999.9221
                </span>
            </div>
            <div class="mt_05 copy_t">
Copyright © 2023 TeenTalk. All rights reserved.
            </div>
        </div>
    </div>
</div>

```

change_img.css

```

@charset "utf-8";
.contentsclear:both; }
.contents::aftercontent:""; clear:both; display:block; width:100%; }
.vsclear:both; width:100%; min-height:400px; background-color: white;
position:relative; overflow: hidden;}

```

```

.img_box position: relative; width: 100%; height: 90vh; overflow: hidden; }
.img_box livisibility: hidden; opacity: 0; transition-duration: 0.7s;
    transition-delay: 0.1s; width: 100%; height: 90vh; }
.img_box li .bg_box width: 100vw; height: 90vh; background-repeat: no-repeat;
    background-position: center center; background-size: cover;
position: absolute;
    left: 0; top: 0; z-index: 5; }
.img_box li.active .bg_box z-index: 6; }
.img_box li .vs_tit position: absolute; top: 180px; left: 100px; z-index: 10;
    font-weight: 300; font-size: 60px; line-height: 1.3; }
.img_box li .vs_tit strong font-weight: 500; }
.img_box li.active visibility: visible; opacity: 1; }
.vs_tit { font-size: 60px; color: white; }
.btn_box li .vs_btn { display: block; width: 12px; height: 12px;
background-color: rgba(255, 255, 255, 0.8);
    border: 2px solid white; position: absolute; top: 100px; left: 100px;
z-index: 14; cursor: pointer;
    border-radius: 50%; }
.btn_box li.item2 .vs_btn { left: 132px; }
.btn_box li.active .vs_btn { background-color: white; border: 2px solid #333; }
.vs_ra { display: none; }
.play_btn { display: block; width: 12px; height: 12px; position: absolute; top:
98px;
    left: 164px; z-index: 14; cursor: pointer; color: white; border: 0;
background-color: transparent;
    font-weight: 900; }
.play_btn:after content: "||"; }
.play_btn.active:after content: "▶"; }

```

common.css

```

@charset "utf-8";
/* 스타일 사용자 정의 초기화 */
body, html {
    width: 100%;
    font-family: 'NanumSquare';
    overflow-x: hidden;
    -ms-overflow-style: none;
}
::-webkit-scrollbar {
    display: none;
}
a {
    text-decoration: none;
    color: #333;
}
img {
    border: 0;
}
.wrap {
    clear: both;
}

```

```

        width: 100%;
    }
    .wrap::after{
        content: "";
        display: block;
        width: 100%;
        clear: both;
    }

```

font.css

```

@font-face{
    font-family: 'NanumSquare';
    src:
url('https://cdn.rawgit.com/moonspam/NanumSquare/master/nanumsquare.css');
    font-weight: normal;
    font-style: normal;
}
@import url('https://cdn.rawgit.com/moonspam/NanumSquare/master/nanumsquare.c
ss');

```

ft.css

```

@charset "utf-8";
.footer_wrap{
    background: #E5E5E5;
    font-size: 0.8125em;
    padding: 2.92308em 0;
    text-align: center;
    box-shadow: 0px -10px 30px #ccc;
}
.footer_wrap .add_bx span:before{
    border-color: #aaa;
}
.footer_wrap .add_bx span:before{
    content: '';
    border-left: 1px solid #666;
    height: 10px;
    position: absolute;
    top: 50%;
    left: -1px;
    margin-top: -5px;
}
.footer_wrap .add_bx span{
    padding: 0 10px;
    position: relative;
    color: #999;
}
.footer_wrap .add_bx span:first-child:before{
    display: none;
}
.add_bx div{

```



```
    display: block;
}
```

hd.css

```
@charset "utf-8";
/* 헤더 구역 스타일 */
.hd{
    clear: both;
    position: relative;
    width: 100%;
    height: 110px;
}
.hd_wrap{
    clear: both;
    width: 100%;
}
.hd_wrap:first-child{
    background-color: #e6ecf6;
    height: 36px;
    /*border-bottom: 1px solid #333;*/
}
.hd_wrap:last-child{
    background-color: #fff;
    width: 1200px;
    margin: 0 auto;
}
.tnb{
    width: 1200px;
    margin: 0 auto;
}
.tnb::after{
    content: "";
    clear: both;
    display: block;
    width: 100%;
}
.tnb .menu{
    float: right;
    height: 36px;
    margin: auto;
    line-height: 36px;
}
.tnb .menu li{
    float: left;
    padding-left: 36px;
    font-size: 16px;
}
.logo{
    display: block;
    width: auto;
```

```
    height: 46px;
    float: left;
}
.logo img{
    display: block;
    width: auto;
    height: 100%;
    margin-top: 12px;
}
.gnb{
    width: 900px;
    margin: 0 auto;
    float: right;
    height: 70px;
}
.gnb::before{
    content: "";
    /*display: block;*/
    width: 100vw;
    height: 180px;
    background-color: rgb(255, 255, 255);
    position: absolute;
    left: 0;
    top: 90px;
    z-index: 15;
    display: none;
}
.gnb:hover::before{
    display: block;
}
.gnb .menu{
    clear: both;
    height: 50px;
    position: relative;
    margin-left: 30px;
}
.gnb .menu::after{
    content: "";
    clear: both;
    width: 100%;
    display: block;
}
.gnb .menu li{
    float: left;
    width: 20%;
    position: relative;
    height: 50px;
    z-index: 9999;
}
.gnb .menu li .sub{
```

```
    visibility: hidden;
    opacity: 0;
    position: absolute;
    width: 180px;
    height: 180px;
    top: 48px;
    left: 0;
    z-index: 999;
    transition: 0.3s;
    margin-top: 20px;
}
.gnb .menu li a.dp1{
    display: block;
    width: 100%;
    line-height: 70px;
    /*border-bottom: 2px solid #fff;*/
    text-align: center;
    background-color: #fff;
    color: #333;
    font-weight: bold;
    font-size: 22px;
}
```

```
.gnb .menu li:hover a.dp1{
    /*font-size: 24px;*/
    border-bottom: 6px solid #BBCDF2;
    /*background-color: #BBCDF2;*/
    line-height: 70px;
}
```

```
.gnb .menu:hover li .sub{
    visibility: visible;
    margin-top: 30px;
    height: 160px;
    opacity: 1;
}
```

```
.gnb .menu li .sub li{
    padding-left: 0px;
    line-height: 34px;
    clear: both;
    width: 180px;
}
```

```
.gnb .menu li .sub li a{
    display: block;
    width: 180px;
    text-align: center;
    line-height: 34px;
    white-space: nowrap;
    text-overflow: ellipsis;
    overflow: hidden;
}
```

```

padding-left: 8px;
padding-right: 8px;
box-sizing: border-box;
font-size: 18px;
color: #7e7e7e;
}
.gnb .menu li .sub li:hover a{
/*font-size: 20px;*/
/*background-color: #BBCDF2;*/
font-weight: bold;
color: #0c0c0c;
}

```

reset.css

```

/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126
License: none (public domain)
*/
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video{
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section{
display: block;
}
body{
line-height: 1;
}
ol, ul{
list-style: none;
}
blockquote, q{

```

```

        quotes: none;
    }
blockquote:before, blockquote:after,
q:before, q:after{
    content: '';
    content: none;
}
table{
    border-collapse: collapse;
    border-spacing: 0;
}

```

roocation.js

```

$(function(){
    ".btn_box li .vs_btn").click(function(){
        varpar = $(this).parents("li").index();
        $(".img_box li").removeClass("active");
        $(".img_box li").eq(par).addClass("active");
        $(".btn_box li").removeClass("active");
        $(".btn_box li").eq(par).addClass("active");
    });
    varsw = 1;
    varint1 = setInterval(function(){
        if(sw==1){
autoplay(1);
sw = 0;
} else{
autoplay(0);
sw = 1;
}
4500);
    functionautoplay(n){
$(".img_box li").removeClass("active");
$(".img_box li").eq(n).addClass("active");
$(".btn_box li").removeClass("active");
$(".btn_box li").eq(n).addClass("active");
}
    ".play_btn").click(function(){
        if($(this).hasClass("active")){
$(this).removeClass("active");
sw = 1;
int1 = setInterval(function(){
            if(sw==1){
autoplay(1);
sw = 0;
} else{
autoplay(0);
sw = 1;
}
}, 3500);

```

```

} else{
$(this).addClass("active");
sw = 0;
clearInterval(int1);
}
});
});

```

2. 모든 화면의 공통 css 코드

common.css

```

@charset "utf-8";
/* 스타일 사용자 정의 초기화 */
body, html{
    width: 100%;
    font-family: 'NanumSquare';
    overflow-x: hidden;
    -ms-overflow-style: none;
}
::-webkit-scrollbar{
    display: none;
}
a{
    text-decoration: none;
    color: #333;
}
img{
    border: 0;
}
.wrap{
    clear: both;
    width: 100%;
}
.wrap::after{
    content: "";
    display: block;
    width: 100%;
    clear: both;
}

```

font.css

```

@font-face{
    font-family: 'NanumSquare';
    src:
url('https://cdn.rawgit.com/moonspam/NanumSquare/master/nanumsquare.css');
    font-weight: normal;
    font-style: normal;
}
@import url('https://cdn.rawgit.com/moonspam/NanumSquare/master/nanumsquare.c
ss');

```

ft.css

```
@charset "utf-8";
.footer_wrap{
    background: #E5E5E5;
    font-size: 0.8125em;
    padding: 2.92308em 0;
    text-align: center;
    box-shadow: 0px -10px 30px #ccc;
}
.footer_wrap .add_bx span:before{
    border-color: #aaa;
}
.footer_wrap .add_bx span:before{
    content: '';
    border-left: 1px solid #666;
    height: 10px;
    position: absolute;
    top: 50%;
    left: -1px;
    margin-top: -5px;
}
.footer_wrap .add_bx span{
    padding: 0 10px;
    position: relative;
    color: #999;
}
.footer_wrap .add_bx span:first-child:before{
    display: none;
}
.add_bx div{
    display: block;
}
```

hd.css

```
@charset "utf-8";
/* 헤더 구역 스타일 */
.hd{
    clear: both;
    position: relative;
    width: 100%;
    height: 110px;
}
.hd_wrap{
    clear: both;
    width: 100%;
}
.hd_wrap:first-child{
    background-color: #e6ecf6;
    height: 36px;
    /*border-bottom: 1px solid #333;*/
}
```



```
}
.hd_wrap:last-child{
    background-color: #fff;
    width: 1200px;
    margin: 0 auto;
}
.tnb{
    width: 1200px;
    margin: 0 auto;
}
.tnb::after{
    content: "";
    clear: both;
    display: block;
    width: 100%;
}
.tnb .menu{
    float: right;
    height: 36px;
    margin: auto;
    line-height: 36px;
}
.tnb .menu li{
    float: left;
    padding-left: 36px;
    font-size: 16px;
}
.logo{
    display: block;
    width: auto;
    height: 46px;
    float: left;
}
.logo img{
    display: block;
    width: auto;
    height: 100%;
    margin-top: 12px;
}
.gnb{
    width: 900px;
    margin: 0 auto;
    float: right;
    height: 70px;
}
.gnb::before{
    content: "";
    /*display: block;*/
    width: 100vw;
    height: 180px;
```

```
background-color: rgb(255, 255, 255);
position: absolute;
left: 0;
top: 90px;
z-index: 15;
display: none;
}
.gnb:hover::before{
display: block;
}
.gnb .menu{
clear: both;
height: 50px;
position: relative;
margin-left: 30px;
}
.gnb .menu::after{
content: "";
clear: both;
width: 100%;
display: block;
}
.gnb .menu li{
float: left;
width: 20%;
position: relative;
height: 50px;
z-index: 9999;
}
.gnb .menu li .sub{
visibility: hidden;
opacity: 0;
position: absolute;
width: 180px;
height: 180px;
top: 48px;
left: 0;
z-index: 999;
transition: 0.3s;
margin-top: 20px;
}
.gnb .menu li a.dp1{
display: block;
width: 100%;
line-height: 70px;
/*border-bottom: 2px solid #fff;*/
text-align: center;
background-color: #fff;
color: #333;
font-weight: bold;
```

```

    font-size: 22px;
}

.gnb .menu li:hover a.dp1{
    /*font-size: 24px;*/
    border-bottom: 6px solid #BBCDF2;
    /*background-color: #BBCDF2;*/
    line-height: 70px;
}

.gnb .menu:hover li .sub{
    visibility: visible;
    margin-top: 30px;
    height: 160px;
    opacity: 1;
}

.gnb .menu li .sub li{
    padding-left: 0px;
    line-height: 34px;
    clear: both;
    width: 180px;
}

.gnb .menu li .sub li a{
    display: block;
    width: 180px;
    text-align: center;
    line-height: 34px;
    white-space: nowrap;
    text-overflow: ellipsis;
    overflow: hidden;
    padding-left: 8px;
    padding-right: 8px;
    box-sizing: border-box;
    font-size: 18px;
    color: #7e7e7e;
}

.gnb .menu li .sub li:hover a{
    /*font-size: 20px;*/
    /*background-color: #BBCDF2;*/
    font-weight: bold;
    color: #0c0c0c;
}

```

reset.css

```

/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126
License: none (public domain)
*/
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,

```

```

a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video{
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section{
    display: block;
}
body{
    line-height: 1;
}
ol, ul{
    list-style: none;
}
blockquote, q{
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after{
    content: '';
    content: none;
}
table{
    border-collapse: collapse;
    border-spacing: 0;
}

```

3. dto 클래스 코드

Cart.java

```

package shop.dto;
public class Cart {
    private int cartno;    //카트번호
    private String cid;    //고객아이디
    private int pno;       //제품번호
    private int amount;    //제품수량
}

```

```

    public int getCartno() {
        return cartno;
    }

    public void setCartno(int cartno) {
        this.cartno = cartno;
    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }

    public int getPno() {
        return pno;
    }

    public void setPno(int pno) {
        this.pno = pno;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }
}

@Override
public String toString() {
    return "Cart{" +
        "cartno=" + cartno +
        ", cid='" + cid + '\'' +
        ", pno=" + pno +
        ", amount=" + amount +
        '}';
}
}

```

cartVO.java

```

package shop.dto;
public class CartVO {
    private int cartno;        //카트번호
    private String cid;        //고객아이디
    private String name;       //고객명
    private int pno;           //제품번호
    private String pname;      //제품명
    private String img1;       //제품 이미지
    private int amount;        //제품수량
    public int getCartno() {
        return cartno;
    }

    public void setCartno(int cartno) {
        this.cartno = cartno;
    }
}

```

```

    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }

    public int getPno() {
        return pno;
    }

    public void setPno(int pno) {
        this.pno = pno;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPname() {
        return pname;
    }

    public void setPname(String pname) {
        this.pname = pname;
    }

    public String getImg1() {
        return img1;
    }

    public void setImg1(String img1) {
        this.img1 = img1;
    }
}

@Override
public String toString() {
    return "CartVO{" +
        "cartno=" + cartno +
        ", cid='" + cid + '\'' +
        ", name='" + name + '\'' +
        ", pno=" + pno +
        ", pname='" + pname + '\'' +
        ", img1='" + img1 + '\'' +
        ", amount=" + amount +
        '}';
}
}

```

category.java

```
packageshop.dto;
//카테고리
public class Category {
    // 초등: 초등, 중등:중등, 고등:고등
    private String cno;
    private String cname; //카테고리명
    public String getCno() {
        return cno;
    }

    public void setCno(String cno) {
        this.cno = cno;
    }

    public String getCname() {
        return cname;
    }

    public void setCname(String cname) {
        this.cname = cname;
    }
    @Override
    public String toString() {
        return "Category{" +
            "cno=" + cno +
            ", cname='" + cname + '\'' +
            '}';
    }
}
```

custom.java

```
packageshop.dto;
public class Custom {
    private String id;
    private String pw;
    private String name;
    private int point;
    private String grade;
    private String tel;
    private String email;
    private String birth;
    private String regdate;
    private String address;
    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getPw() {
        return pw;
    }
}
```



```
    public void setPw(String pw) {
        this.pw = pw;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getPoint() {
        return point;
    }

    public void setPoint(int point) {
        this.point = point;
    }

    public String getGrade() {
        return grade;
    }

    public void setGrade(String grade) {
        this.grade = grade;
    }

    public String getTel() {
        return tel;
    }

    public void setTel(String tel) {
        this.tel = tel;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getBirth() {
        return birth;
    }

    public void setBirth(String birth) {
        this.birth = birth;
    }

    public String getRegdate() {
        return regdate;
    }

    public void setRegdate(String regdate) {
        this.regdate = regdate;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }
}
```

```

}
@Override
public String toString() {
    return "Custom{" +
        "id='" + id + '\'' +
        ", pw='" + pw + '\'' +
        ", name='" + name + '\'' +
        ", point=" + point +
        ", grade='" + grade + '\'' +
        ", tel='" + tel + '\'' +
        ", email='" + email + '\'' +
        ", birth='" + birth + '\'' +
        ", regdate='" + regdate + '\'' +
        ", address='" + address + '\'' +
        '}';
}
}

```

delivery.java

```

package shop.dto;
public class Delivery {
    private int dno;    //(auto)배송코드
    private int sno;    //결제번호
    private String cid; //고객아이디
    private String daddr; //배송지
    private String custel; //고객연락처
    private String pcom; //(admin)배송회사
    private String ptel; //(admin)배송기사연락처
    private int pstate; //(0:배송전,1:배송중,2:도착,3:구매결정)-배송상태
    private String sdate;
    private String rdate;
    private String bcode; //화물코드
    public int getDno() {
        return dno;
    }
    public void setDno(int dno) {
        this.dno = dno;
    }
    public String getSdate() {
        return sdate;
    }
    public void setSdate(String sdate) {
        this.sdate = sdate;
    }
    public String getRdate() {
        return rdate;
    }
    public void setRdate(String rdate) {
        this.rdate = rdate;
    }
}

```

```

}

    public int getSno() {
        return sno;
    }

    public void setSno(int sno) {
        this.sno = sno;
    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }

    public String getDaddr() {
        return daddr;
    }

    public void setDaddr(String daddr) {
        this.daddr = daddr;
    }

    public String getCustel() {
        return custel;
    }

    public void setCustel(String custel) {
        this.custel = custel;
    }

    public String getPcom() {
        return pcom;
    }

    public void setPcom(String pcom) {
        this.pcom = pcom;
    }

    public String getPtel() {
        return ptel;
    }

    public void setPtel(String ptel) {
        this.ptel = ptel;
    }

    public int getPstate() {
        return pstate;
    }

    public void setPstate(int pstate) {
        this.pstate = pstate;
    }

    public String getBcode() {
        return bcode;
    }

    public void setBcode(String bcode) {
        this.bcode = bcode;
    }
}
@Override

```

```

public String toString() {
    return "Delivery{"+
        "dno="+ dno +
        ", sno="+ sno +
        ", cid='"+ cid + '\''+
        ", daddr='"+ daddr + '\''+
        ", custel='"+ custel + '\''+
        ", pcom='"+ pcom + '\''+
        ", ptel='"+ ptel + '\''+
        ", pstate="+ pstate +
        ", sdate="+ sdate +
        ", rdate="+ rdate +
        ", bcode='"+ bcode + '\''+
        '}';
}
}

```

deliveryInfoView.java

```

packageshop.dto;
public class DeliveryInfoView {
    private int dno;    //(auto)배송코드
    private int sno;    //결제번호
    private String cid; //고객아이디
    private String daddr; //+배송지
    private String custel; //+고객연락처
    private String pcom; //(admin)배송회사
    private String ptel; //(admin)배송기사연락처
    private int pstate; //(0:배송전,1:배송중,2:도착,3:구매결정)-배송상태
    private String sdate; // 결제일
    private String rdate; // 배송시작일
    private String bcode; //운송 번호
    private String pname; // 제품 번호
    private int amount; // 구매 수량
    public int getDno() {
        return dno;
    }

    public void setDno(int dno) {
        this.dno = dno;
    }

    public int getSno() {
        return sno;
    }

    public void setSno(int sno) {
        this.sno = sno;
    }

    public String getCid() {
        return cid;
    }
}

```

```
    public void setCid(String cid) {
        this.cid = cid;
    }

    public String getDaddr() {
        return daddr;
    }

    public void setDaddr(String daddr) {
        this.daddr = daddr;
    }

    public String getCustel() {
        return custel;
    }

    public void setCustel(String custel) {
        this.custel = custel;
    }

    public String getPcom() {
        return pcom;
    }

    public void setPcom(String pcom) {
        this.pcom = pcom;
    }

    public String getPtel() {
        return ptel;
    }

    public void setPtel(String ptel) {
        this.ptel = ptel;
    }

    public int getPstate() {
        return pstate;
    }

    public void setPstate(int pstate) {
        this.pstate = pstate;
    }

    public String getSdate() {
        return sdate;
    }

    public void setSdate(String sdate) {
        this.sdate = sdate;
    }

    public String getRdate() {
        return rdate;
    }

    public void setRdate(String rdate) {
        this.rdate = rdate;
    }

    public String getBcode() {
        return bcode;
    }

    public void setBcode(String bcode) {
        this.bcode = bcode;
    }
}
```

```

    }

    public String getPname() {
        return pname;
    }

    public void setPname(String pname) {
        this.pname = pname;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }
}

@Override
public String toString() {
    return "DeliveryInfoView{" +
        "dno=" + dno +
        ", sno=" + sno +
        ", cid='" + cid + '\'' +
        ", daddr='" + daddr + '\'' +
        ", custel='" + custel + '\'' +
        ", pcom='" + pcom + '\'' +
        ", ptel='" + ptel + '\'' +
        ", pstate=" + pstate +
        ", sdate='" + sdate + '\'' +
        ", rdate='" + rdate + '\'' +
        ", bcode='" + bcode + '\'' +
        ", pname='" + pname + '\'' +
        ", amount='" + amount + '\'' +
        '}';
}
}

```

notice.java

```

package shop.dto;
public class Notice {
    private int no;
    private String title;
    private String content;
    private String resdate;
    private int visited;
    public int getNo() {
        return no;
    }

    public void setNo(int no) {
        this.no = no;
    }

    public String getTitle() {
        return title;
    }
}

```

```

    public void setTitle(String title) {
        this.title = title;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }

    public String getResdate() {
        return resdate;
    }

    public void setResdate(String resdate) {
        this.resdate = resdate;
    }

    public int getVisited() {
        return visited;
    }

    public void setVisited(int visited) {
        this.visited = visited;
    }
}

@Override
public String toString() {
    return "Notice{" +
        "no=" + no +
        ", title='" + title + '\'' +
        ", content='" + content + '\'' +
        ", resdate='" + resdate + '\'' +
        ", visited=" + visited +
        '}';
}
}

```

payment.java

```

package shop.dto;
public class Payment {
    private int sno;           //(auto)결제번호
    private String cid;        //고객아이디
    private int pno;           //상품번호
    private int amount;        //구매수량
    private String pmethod;    //+결제수단
    private String pcom;       //+결제 대행사
    private String pnum;       //+결제카드(계좌)번호
    private int payprice;      //+결제금액
    private String dno;        //배송코드
    public int getSno() {
        return sno;
    }

    public void setSno(int sno) {
        this.sno = sno;
    }
}

```



```

}

    public int getPno() {
        return pno;
    }

    public void setPno(int pno) {
        this.pno = pno;
    }

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }

    public String getDno() {
        return dno;
    }

    public void setDno(String dno) {
        this.dno = dno;
    }

    public String getPmethod() {
        return pmethod;
    }

    public void setPmethod(String pmethod) {
        this.pmethod = pmethod;
    }

    public String getPcom() {
        return pcom;
    }

    public void setPcom(String pcom) {
        this.pcom = pcom;
    }

    public String getCnum() {
        return cnum;
    }

    public void setCnum(String cnum) {
        this.cnum = cnum;
    }

    public int getPayprice() {
        return payprice;
    }

    public void setPayprice(int payprice) {
        this.payprice = payprice;
    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }
}
@Override

```

```

public String toString() {
    return "Payment{" +
        "sno=" + sno +
        ", cid='" + cid + '\'' +
        ", pno=" + pno +
        ", amount=" + amount +
        ", pmethod='" + pmethod + '\'' +
        ", pcom='" + pcom + '\'' +
        ", cnum='" + cnum + '\'' +
        ", payprice=" + payprice +
        ", dno=" + dno +
        '}';
}
}

```

paymentVO.java

```

package shop.dto;
public class PaymentVO {
    private int sno;           //(auto)결제번호
    private String cid;        //고객아이디
    private int pno;           //상품번호
    private int amount;        //구매수량
    private String pmethod;    //결제수단
    private String pcom;       //결제 대행사
    private String cnum;       //결제카드(계좌)번호
    private int payprice;      //결제금액
    private String dno;        //배송코드
    private String pname;      //상품명
    private int pstate;        //배송 상태
    public int getSno() {
        return sno;
    }
    public void setSno(int sno) {
        this.sno = sno;
    }
    public int getPno() {
        return pno;
    }
    public void setPno(int pno) {
        this.pno = pno;
    }
    public int getAmount() {
        return amount;
    }
    public void setAmount(int amount) {
        this.amount = amount;
    }
    public String getDno() {
        return dno;
    }
}

```

```

    public void setDno(String dno) {
        this.dno = dno;
    }

    public String getPmethod() {
        return pmethod;
    }

    public void setPmethod(String pmethod) {
        this.pmethod = pmethod;
    }

    public String getPcom() {
        return pcom;
    }

    public void setPcom(String pcom) {
        this.pcom = pcom;
    }

    public String getCnum() {
        return cnum;
    }

    public void setCnum(String cnum) {
        this.cnum = cnum;
    }

    public int getPayprice() {
        return payprice;
    }

    public void setPayprice(int payprice) {
        this.payprice = payprice;
    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }

    public String getPname() {
        return pname;
    }

    public void setPname(String pname) {
        this.pname = pname;
    }

    public int getPstate() {
        return pstate;
    }

    public void setPstate(int pstate) {
        this.pstate = pstate;
    }
}

@Override
public String toString() {
    return "PaymentVO{" +
        "sno=" + sno +
        ", cid='" + cid + '\'' +

```

```

        ", pno="+ pno +
        ", amount="+ amount +
        ", pmethod='"+ pmethod + '\\' +
        ", pcom='"+ pcom + '\\' +
        ", cnum='"+ cnum + '\\' +
        ", payprice="+ payprice +
        ", dno='"+ dno + '\\' +
        ", pname='"+ pname + '\\' +
        ", pstate="+ pstate +
        '}}';
    }
}

```

product.java

```

package shop.dto;
// 상품
public class Product {
    private int no; // 상품번호
    private String cate; // 카테고리
    private String cateno; // 카테고리번호
    private String pname; // 상품명
    private String pcomment; // 상품설명
    private String plist; // 상품목차
    private int pqty; // 상품 수량
    private int price; // 상품단가
    private String imgSrc1; // 상품이미지
    private String imgSrc2; // 상품이미지
    private String imgSrc3; // 상품이미지
    private String resdate; // 등록일
    public int getNo() {
        return no;
    }
    public void setNo(int no) {
        this.no = no;
    }
    public String getCate() {
        return cate;
    }
    public void setCate(String cate) {
        this.cate = cate;
    }
    public String getCateno() {
        return cateno;
    }
    public void setCateno(String cateno) {
        this.cateno = cateno;
    }
    public int getPqty() {
        return pqty;
    }
}

```

```
public void setPqty(int pqty) {
    this.pqty = pqty;
}

public String getPname() {
    return pname;
}

public void setPname(String pname) {
    this.pname = pname;
}

public String getPcomment() {
    return pcomment;
}

public void setPcomment(String pcomment) {
    this.pcomment = pcomment;
}

public String getPlist() {
    return plist;
}

public void setPlist(String plist) {
    this.plist = plist;
}

public int getPrice() {
    return price;
}

public void setPrice(int price) {
    this.price = price;
}

public String getImgSrc1() {
    return imgSrc1;
}

public void setImgSrc1(String imgSrc1) {
    this.imgSrc1 = imgSrc1;
}

public String getImgSrc2() {
    return imgSrc2;
}

public void setImgSrc2(String imgSrc2) {
    this.imgSrc2 = imgSrc2;
}

public String getImgSrc3() {
    return imgSrc3;
}

public void setImgSrc3(String imgSrc3) {
    this.imgSrc3 = imgSrc3;
}

public String getResdate() {
    return resdate;
}

public void setResdate(String resdate) {
    this.resdate = resdate;
}
```

```

}
@Override
    public String toString() {
        return "Product{" +
            "no=" + no +
            ", cate='" + cate + '\'' +
            ", cateno='" + cateno + '\'' +
            ", pname='" + pname + '\'' +
            ", pcomment='" + pcomment + '\'' +
            ", plist='" + plist + '\'' +
            ", price=" + price +
            ", imgSrc1='" + imgSrc1 + '\'' +
            ", imgSrc2='" + imgSrc2 + '\'' +
            ", imgSrc3='" + imgSrc3 + '\'' +
            ", resdate='" + resdate + '\'' +
            '}';
    }
}

```

Qna.java

```

package shop.dto;
public class Qna {
    private int qno;
    private String title;
    private String content;
    private String cid;
    private String resdate;
    private int cnt;
    private int lev;
    private int par;
    public int getQno() {
        return qno;
    }
    public void setQno(int qno) {
        this.qno = qno;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getContent() {
        return content;
    }
    public void setContent(String content) {
        this.content = content;
    }
    public String getCid() {
        return cid;
    }
}

```

```

    }

    public void setCid(String cid) {
        this.cid = cid;
    }

    public String getResdate() {
        return resdate;
    }

    public void setResdate(String resdate) {
        this.resdate = resdate;
    }

    public int getCnt() {
        return cnt;
    }

    public void setCnt(int cnt) {
        this.cnt = cnt;
    }

    public int getLev() {
        return lev;
    }

    public void setLev(int lev) {
        this.lev = lev;
    }

    public int getPar() {
        return par;
    }

    public void setPar(int par) {
        this.par = par;
    }
}

@Override
public String toString() {
    return "Qna{" +
        "qno=" + qno +
        ", title='" + title + '\'' +
        ", content='" + content + '\'' +
        ", cid='" + cid + '\'' +
        ", resdate='" + resdate + '\'' +
        ", cnt=" + cnt +
        ", lev=" + lev +
        ", par=" + par +
        '}';
}
}

```

Receive.java

```

package shop.dto;

public class Receive {
    private int rno;           //입고번호(auto)
    private int pno;           //입고 제품번호
    private int amount;        //입고수량
    private int rprice;        //입고가격
}

```

```

    private String resdate; //입고일
    public int getRno() {
        return rno;
    }
    public void setRno(int rno) {
        this.rno = rno;
    }
    public int getPno() {
        return pno;
    }
    public void setPno(int pno) {
        this.pno = pno;
    }
    public int getAmount() {
        return amount;
    }
    public void setAmount(int amount) {
        this.amount = amount;
    }
    public int getRprice() {
        return rprice;
    }
    public void setRprice(int rprice) {
        this.rprice = rprice;
    }
    public String getResdate() {
        return resdate;
    }
    public void setResdate(String resdate) {
        this.resdate = resdate;
    }
    @Override
    public String toString() {
        return "Receive{" +
            "rno=" + rno +
            ", pno=" + pno +
            ", amount=" + amount +
            ", rprice=" + rprice +
            ", resdate='" + resdate + '\'' +
            '}';
    }
}

```

Review.java

```

package shop.dto;
public class Review {
    private int no;
    private String cid;
    private String content;
    private int par;
}

```



```

    private Stringpname;
    private Stringresdate;
    public intgetNo() {
        returnno;
    }
    public voidsetNo(intno) {
        this.no = no;
    }
    public StringgetCid() {
        returncid;
    }
    public voidsetCid(Stringcid) {
        this.cid = cid;
    }
    public StringgetContent() {
        returncontent;
    }
    public voidsetContent(Stringcontent) {
        this.content = content;
    }
    public intgetPar() {
        returnpar;
    }
    public voidsetPar(intpar) {
        this.par = par;
    }
    public StringgetResdate() {
        returnresdate;
    }
    public voidsetResdate(Stringresdate) {
        this.resdate = resdate;
    }
    public StringgetPname() {
        returnpname;
    }
    public voidsetPname(Stringpname) {
        this.pname = pname;
    }
    @Override
    public StringtoString() {
        return "Review{"+
            "no="+ no +
            ", cid='"+ cid + '\''+
            ", content='"+ content + '\''+
            ", par="+ par +
            ", resdate='"+ resdate + '\''+
            '}';
    }
}

```

sales.java

```
package shop.dto;
public class Sales {
    private int pno;
    private String pname;
    private int cnt;
    private int money;
    public int getPno() {
        return pno;
    }
    public void setPno(int pno) {
        this.pno = pno;
    }
    public String getPname() {
        return pname;
    }
    public void setPname(String pname) {
        this.pname = pname;
    }
    public int getCnt() {
        return cnt;
    }
    public void setCnt(int cnt) {
        this.cnt = cnt;
    }
    public int getMoney() {
        return money;
    }
    public void setMoney(int money) {
        this.money = money;
    }
    @Override
    public String toString() {
        return "Sales{" +
            "pno=" + pno +
            ", pname='" + pname + '\'' +
            ", cnt=" + cnt +
            ", money=" + money +
            '}';
    }
}
```

serve.java

```
package shop.dto;
public class Sales {
    private int pno;
    private String pname;
    private int cnt;
    private int money;
    public int getPno() {
```

```

        return pno;
    }

    public void setPno(int pno) {
        this.pno = pno;
    }

    public String getPname() {
        return pname;
    }

    public void setPname(String pname) {
        this.pname = pname;
    }

    public int getCnt() {
        return cnt;
    }

    public void setCnt(int cnt) {
        this.cnt = cnt;
    }

    public int getMoney() {
        return money;
    }

    public void setMoney(int money) {
        this.money = money;
    }
}

@Override
public String toString() {
    return "Sales{" +
        "pno=" + pno +
        ", pname='" + pname + '\'' +
        ", cnt=" + cnt +
        ", money=" + money +
        '}';
}
}

```

4. 유틸리티에 해당하는 코드

aes256.java

```

package shop.util;
import javax.crypto.*;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import java.io.UnsupportedEncodingException;
import java.nio.ByteBuffer;
import java.security.*;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.InvalidParameterSpecException;
import java.util.Base64;
public class AES256 {

```

```

/**
 * MD5 로 해시 한다.
 *
 * @parammsg
 * @return
 * public static Stringmd5(Stringmsg) throws NoSuchAlgorithmException{
 *     MessageDigestmd = MessageDigest.getInstance("MD5");
md.update(msg.getBytes());
    returnAES256.byteToHexString(md.digest());
}
/**
 * SHA-256으로 해시 한다.
 *
 * @parammsg
 * @return
 * public static Stringsha256(Stringmsg) throws NoSuchAlgorithmException{
 *     MessageDigestmd = MessageDigest.getInstance("SHA-256");
md.update(msg.getBytes());
    returnAES256.byteToHexString(md.digest());
}
/**
 * 바이트 배열을 HEX 문자열로 변환한다.
 * @paramdata
 * @return
 * public static StringbyteToHexString(byte[] data) {
 *     StringBuildersb = newStringBuilder();
 *     for(byte:data) {
sb.append(Integer.toString((b & 0xff) + 0x100, 16).substring(1));
 *     }
 *     returnsb.toString();
 * }
/**
 * AES 256 으로 암호화 한다.
 * @parammsg
 * @paramkey
 * @return
 * @throwsNoSuchAlgorithmException
 * @throwsInvalidKeySpecException
 * @throwsNoSuchPaddingException
 * @throwsInvalidKeyException
 * @throwsInvalidParameterSpecException
 * @throwsUnsupportedEncodingException
 * @throwsBadPaddingException
 * @throwsIllegalBlockSizeException
 * public static StringencryptAES256(Stringmsg, Stringkey) throws
NoSuchAlgorithmException, InvalidKeySpecException, NoSuchPaddingException,
InvalidKeyException, InvalidParameterSpecException,
UnsupportedEncodingException, BadPaddingException, IllegalBlockSizeException{
 *     SecureRandomrandom = newSecureRandom();
 *     bytebytes[] = new byte[20];

```

```

random.nextBytes(bytes);
    byte[] saltBytes = bytes;
    // Password-Based Key Derivation function 2
    SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
    // 70000번 해시하여 256 bit 길이의 키를 만든다.
    PBEKeySpec spec = new PBEKeySpec(key.toCharArray(), saltBytes, 70000,
256);
    SecretKey secretKey = factory.generateSecret(spec);
    SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(),
"AES");
    // 알고리즘/모드/패딩
    // CBC : Cipher Block Chaining Mode
    Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE, secret);
    AlgorithmParameters params = cipher.getParameters();
    // Initial Vector(1단계 암호화 블록용)
    byte[] ivBytes =
params.getParameterSpec(IvParameterSpec.class).getIV();
    byte[] encryptedTextBytes = cipher.doFinal(msg.getBytes("UTF-8"));
    byte[] buffer = new byte[saltBytes.length + ivBytes.length +
encryptedTextBytes.length];
    System.arraycopy(saltBytes, 0, buffer, 0, saltBytes.length);
    System.arraycopy(ivBytes, 0, buffer, saltBytes.length, ivBytes.length);
    System.arraycopy(encryptedTextBytes, 0, buffer, saltBytes.length +
ivBytes.length, encryptedTextBytes.length);
    return Base64.getEncoder().encodeToString(buffer);
}
/**
 * 위에서 암호화된 내용을 복호화 한다.
 * @param msg
 * @param key
 * @return
 * @throws NoSuchPaddingException
 * @throws NoSuchAlgorithmException
 * @throws InvalidKeySpecException
 * @throws InvalidAlgorithmParameterException
 * @throws InvalidKeyException
 * @throws BadPaddingException
 * @throws IllegalBlockSizeException
    public static String decryptAES256(String msg, String key) throws
NoSuchPaddingException, NoSuchAlgorithmException, InvalidKeySpecException,
InvalidAlgorithmParameterException, InvalidKeyException, BadPaddingException,
IllegalBlockSizeException{
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        ByteBuffer buffer = ByteBuffer.wrap(Base64.getDecoder().decode(msg));
        byte[] saltBytes = new byte[20];
buffer.get(saltBytes, 0, saltBytes.length);
        byte[] ivBytes = new byte[cipher.getBlockSize()];
buffer.get(ivBytes, 0, ivBytes.length);

```

```

        byte[] encryptedTextBytes = new byte[buffer.capacity() -
saltBytes.length - ivBytes.length];
buffer.get(encryptedTextBytes);

        SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
        PBEKeySpec spec = new PBEKeySpec(key.toCharArray(), saltBytes, 70000,
256);
        SecretKey secretKey = factory.generateSecret(spec);
        SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(),
"AES");
cipher.init(Cipher.DECRYPT_MODE, secret, new IvParameterSpec(ivBytes));
        byte[] decryptedTextBytes = cipher.doFinal(encryptedTextBytes);
        return new String(decryptedTextBytes);
    }
}

```

sha256.java

```

package com.chunjae.util;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class SHA256 {
    public static String encrypt(String text) throws NoSuchAlgorithmException {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
md.update(text.getBytes());
        return bytesToHex(md.digest());
    }

    private static String bytesToHex(byte[] bytes) {
        StringBuilder sb = new StringBuilder();
        for (byte b : bytes) {
sb.append(String.format("%02x", b));
        }

        return sb.toString();
    }
}

```

SMTPAuthentication.java

```

package com.chunjae.util;
import com.chunjae.data.Email;
import javax.mail.Authenticator;
import javax.mail.PasswordAuthentication;
// 사용할 메일과 비밀번호 설정 가능
// lib에 넣는 경우 동작에 문제는 없지만 인텔리제이 상에서 빨간 줄이 뜬다.
public class SMTPAuthenticator extends Authenticator {
@Override
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(Email.ID, Email.PW);
    }
}

```

```
}  
}
```

sendMail.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"  
language="java" %>  
<!-- 1. 라이브러리 가져옴 -->  
<%@ page import="com.chunjae.util.*" %>  
<%@ page import="java.util.*, javax.mail.*, javax.mail.internet.*" %>  
<!-- 2. 인코딩 설정 -->  
<%@ include file="/encoding.jsp"%>  
<!-- 학원에서는 안되니 집에서 해봐라 -->  
<%  
//3. 보내온 이메일 데이터 받기  
String name = request.getParameter("name");  
String id = request.getParameter("id");  
String email = request.getParameter("email"); //보내는 사람 이메일  
String tel = request.getParameter("tel");  
String title = request.getParameter("title");  
String content = request.getParameter("content");  
String to = "do11anm@naver.com"; //받는 사람 이메일  
//4. smtp 설정 -> Properties 활용  
Properties p = new Properties();  
p.put("mail.smtp.host", "smtp.naver.com");  
p.put("mail.smtp.port", "465");  
p.put("mail.smtp.starttls.enable", "true");  
p.put("mail.smtp.ssl.enable", "true");  
p.put("mail.smtp.auth", "true");  
p.put("mail.smtp.debug", "true");  
p.put("mail.smtp.socketFactory.port", "465");  
p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");  
p.put("mail.smtp.socketFactory.fallback", "false");  
//5. 이메일 보내기  
try {  
Authenticator auth = new SMTPAuthenticator();  
Session ses = Session.getInstance(p, auth); // 인증 권한이 필요할 때 메일을 보낼  
경우  
// Session.getDefaultInstance(p) //인증 권한이 필요 없을 때 이메일을 보낼  
경우 사용, 이미 로그인 된 경우  
ses.setDebug(true);  
MimeMessage msg = new MimeMessage(ses); // 파일 첨부나 객체 까지 전달할 때 사용  
하는 클래스  
msg.setSubject(title); // 제목 추가  
StringBuffer buffer = new StringBuffer(); //이메일 내용  
buffer.append("보내는 사람 : ");  
buffer.append(name+"\n");  
buffer.append("연락처 : ");  
buffer.append(tel+"\n");  
buffer.append("이메일 : ");  
buffer.append(email+"\n");
```

```

buffer.append("제목 : ");
buffer.append(title+"\n");
buffer.append("내용 : ");
buffer.append(content+"\n");
// string을 인터넷 주소로 변형
Address fromAddr = new InternetAddress(email); //보내는 사람
msg.setFrom(fromAddr);
Address toAddr = new InternetAddress(to); // 받는 사람
msg.addRecipient(Message.RecipientType.TO, toAddr); // 하나의 이메일 주소만 등록
// 버퍼의 내용을 보낼때 사용
msg.setContent(buffer.toString(), "text/html;charset=UTF-8"); // 내용
Transport.send(msg); // 메일 전송
} catch (Exception e){
e.printStackTrace();
return;
} finally {
response.sendRedirect("/index.jsp");
}
%>

```

pasing.java

```

package com.chunjae.dto;
public class Pasing {
    private int wSize; // 글 개수
    private int pSize; // 페이지 개수
    private int writingCount;
    private int curPage;
    public Pasing(int wSize, int pSize, int writingCount, int curPage) {
        this.wSize = wSize;
        this.pSize = pSize;
        this.writingCount = writingCount;
        this.curPage = curPage;
    }

    public int getPageCount()
    {
        return (writingCount - 1) / wSize + 1;
    }

    public int getPageStart()
    {
        return (curPage - 1) / pSize * pSize + 1;
    }

    public int getPageEnd()
    {
        return Math.min( getPageStart() + pSize - 1, getPageCount() );
    }

    public boolean isPre()
    {
        return getPageStart() != 1;
    }

    public boolean isNext()

```



```

{
    return getPageEnd() < getPageCount();
}

public int getWritingStart()
{
    return getWritingEnd() - wSize + 1;
}

public int getWritingEnd()
{
    return curPage * wSize;
}
}

```

faqList.jsp

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="com.chunjae.db.*" %>
<%@ page import="com.chunjae.vo.*" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="com.chunjae.dto.Faq" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>QNA</title>
<%@ include file="../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결 -->
    <!-- jquery -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- datatables -->
    <link
rel="stylesheet" href="https://cdn.datatables.net/1.13.6/css/jquery.dataTables
s.css">
    <script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet" href="../css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet" href="../css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet" href="../css/common.css">

```

```

<link rel="stylesheet"href="../css/hd.css">
<link rel="stylesheet"href="../css/ft.css">
<style>
    /* 본문 영역 스타일 */
        .contentsclear:both; min-height: 100vh; background-image:
url("../img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
    }

    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 100vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
        .page_titfont-size:48px; text-align: center; padding-top:0.7em;
color:#fff;
        padding-bottom: 1.3em; }
    .breadcrumbclear:both;
        width:1200px; margin: 0 auto; text-align: right; color:#fff;
        padding-top: 28px; padding-bottom: 28px; }
    .breadcrumb acolor:#fff; }
    .tblwidth:600px; margin:0 auto; font-size: 24px;}
    .tbl th{line-height: 32px; padding-top:16px; padding-bottom:16px;
        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
    .tbl td{line-height: 32px; padding-top:16px; padding-bottom:16px;
        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
    .tbl .item1width: 10%;}
    .tbl .item2{width: 60%;}
    .tbl .item3{width: 10%;}
    .tbl .item4{width: 20%;}
    .inbtndisplay:block; border-radius:100px;
        min-width:140px; padding-left: 24px; padding-right: 24px;
text-align: center;
        line-height: 48px; background-color: #333; color:#fff; font-size:
18px; float:right;
        cursor: pointer;}
    .inbtn:hover{
        background-color: #666666;
    }

    .btn_group2{display: none; margin-top: 5px;}
        .inbtn2{background-color: #333; color: white; border-radius: 5px;
padding: 5px; margin-right: 30px;}
        .inbtn2:hover{
            background-color: #666666;
        }

    #delete_btn{background-color: red;}
    #delete_btn:hover{background-color: brown;}
    .ans{display: none;}
    .faqlist{font-size: 20px; }
    .que{cursor: pointer;}

```

```

        .paginationfont-size: 18px; text-align: center; }
        .paging{margin-right: 10px; margin-bottom: 10px;}
    </style>
</head>
<%
request.setCharacterEncoding("utf8");
response.setContentType("text/html;charset=UTF-8");
response.setCharacterEncoding("utf8");
int itemsPerPage = 10; // 한 페이지에 보여줄 아이템 수
int    currentPage    =    (request.getParameter("page")    !=    null)    ?
Integer.parseInt(request.getParameter("page")) : 1;
// faqList 전체 데이터와 총 개수를 가정하고 나눠주는 로직
List<Faq>faqList = new ArrayList<>(); // faqList는 예시를 위한 가상의 데이터라
가정합니다.
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;
DBC con = new MariaDBCon();
conn = con.connect();
if(conn != null){
System.out.println("DB 연결 성공");
}
// 해당 회원의 정보를 db에서 가져옴
try {
String sql = "select * from faq order by fno desc";
pstmt = conn.prepareStatement(sql);
rs = pstmt.executeQuery();
while(rs.next()){
Faq faq = new Faq();
faq.setFno(rs.getInt("fno"));
faq.setQuestion(rs.getString("question"));
faq.setAnswer(rs.getString("answer"));
faq.setCnt(rs.getInt("cnt"));
faqList.add(faq);
}
} catch(SQLException e) {
System.out.println("SQL 구문이 처리되지 못했습니다.");
} finally {
con.close(rs, pstmt, conn);
}
%>
<body>
<div class="wrap">
    <header class="hd" id="hd">
<%@ include file="../header.jsp" %>
    </header>
    <div class="contents" id="contents">
        <div class="breadcrumb">
            <p><a href="/>HOME</a>>
            <a href="/qna/qnaList.jsp">FAQ</a></p>

```

```

</div>
<section class="page" id="page1">
    <div class="page_wrap">
        <h2 class="page_tit">FAQ<%= " p-" + currentPage %></h2>
        <ul class="faqlist">
<!-- 페이징 처리된 faq 출력 -->
<%
int totalItems = faqList.size();
int totalPages = (int) Math.ceil((double) totalItems / itemsPerPage);
int startItem = (currentPage - 1) * itemsPerPage;
int endItem = Math.min(startItem + itemsPerPage, totalItems);
for (int i = startItem; i < endItem; i++) {
    Faq f = faqList.get(i);
    System.out.println(faqList.get(i));
    %>
        <li>
            <div class="que"><%= f.getQuestion() %></div>
            <br>
            <div class="ans"><%= f.getAnswer() %></div>
            <div class="btn_group2">
<% if (sid != null && sid.equals("admin")) { %>
                <a href="/faq/updatefaq.jsp?fno=<%= f.getFno()
%>&page=<%= currentPage %>" class="inbtn2">글 수정</a>
                <a href="/faq/deletefaqpro.jsp?fno=<%= f.getFno()
%>&page=<%= currentPage %>" class="inbtn2" id="delete_btn">글 삭제</a>
            <% } %>
            </div>
        </li>
        <hr>
    <%
    }
    %>
    </ul>
<!-- 페이지 네비게이션 -->
    <div class="pagination">
<% if (currentPage > 1) { %>
        <a class="paging" href="?page=<%= currentPage - 1
%>">Previous</a>
    <% } %>
    <% for (int i = 1; i <= totalPages; i++) { %>
        <a class="paging" href="?page=<%= i %>"><%= i %></a>
    <% } %>
    <% if (currentPage < totalPages) { %>
        <a class="paging" href="?page=<%= currentPage + 1
%>">Next</a>
    <% } %>
    </div>
<!-- 제이쿼리를 이용하여 FAQ 게시판 만들기,
.faglist 내의 li 태그를 누를 경우 해당 질문이 보이게 된다. -->
<script>

```

```
$(document).ready(function() {
$(".faqlist li").click(function() {
$(this).find(".ans").slideToggle(500);
$(this).find(".btn_group2").slideToggle(500);
});
});

</script>
<div class="btn_group">
<% if (sid != null && sid.equals("admin")) { %>
<a href="/faq/addFaq.jsp" class="inbtn">faq 작성 </a>
<p class="exp">관리자만 댓글을 작성 할 수 있습니다.</p>
<% } %>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="../footer.jsp"%>
</footer>
</div>
</body>
</html>
```

해당 능력단위 요소	처리 기능 구현하기	(20점)
수행3. 작성 내용		

1. 회원에 관한 컨트롤러

idcheck.java

```

package shop.controller.custom;
import org.json.JSONObject;
import shop.model.CustomDAO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/IdCheck.do")
public class IdCheckCtrl extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException{
        String id = req.getParameter("id");
        CustomDAO dao = new CustomDAO();
        boolean pass = false;
        pass = dao.idCheck(id);
        JSONObject json = new JSONObject();
        json.put("result", pass);
        PrintWriter out = resp.getWriter();
        out.println(json.toString());
        /*
        String id = req.getParameter("id");
        CustomDAO dao = new CustomDAO();
        boolean idExists = dao.idCheck(id);
        System.out.println(id);
        System.out.println(idExists);
        JSONObject jsonResponse = new JSONObject();
        jsonResponse.put("result", idExists);
        resp.setContentType("application/x-json; charset=utf-8");
        resp.getWriter().write(jsonResponse.toString()); // 전송이 되는 부분*/
    }
}

```

joinctrl.java

```
package shop.controller.custom;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/Join.do")
public class JoinCtrl extends HttpServlet {
    @Override
        protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        request.setAttribute("msg", "회원가입 하시기 바랍니다.");
        RequestDispatcher view =
        request.getRequestDispatcher("/custom/join.jsp");
        view.forward(request, response);
    }
}
```

joinproctrl.java

```
package shop.controller.custom;
import shop.model.CustomDAO;
import shop.util.AES256;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/JoinPro.do")
public class JoinProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        String id = request.getParameter("id");
        String pw = request.getParameter("pw");
        String name = request.getParameter("name");
        String tel = request.getParameter("tel");
        String email = request.getParameter("email");
        String birth = request.getParameter("birth");
        String address1 = request.getParameter("address1");
        String address2 = request.getParameter("address2");
        String postcode = request.getParameter("postcode");
        String msg = "회원가입 페이지";
        CustomDAO dao = new CustomDAO();
```

```

System.out.printf("%s %s %s %s %s %s %s ", id, pw, name, tel, email, birth,
(address1 + "<br>" + address2));
    Stringkey = "%02x";
    Stringencrypted = "";
    try{
encrypted = AES256.encryptAES256(pw, key);
System.out.println("비밀번호 암호화 : "+encrypted);
    } catch(Exceptione) {
e.printStackTrace();
    }

    booleanpass = dao.join(id, encrypted, name, tel, email, birth,
(address1 + "<br>" + address2 + "<br>" + postcode));
System.out.println(pass);
    HttpSessionsession = request.getSession();
    PrintWriterout = response.getWriter();
    RequestDispatcherview = null;
    if(pass) {
msg = "회원가입 성공";
//        session.setAttribute("sid", id);
//        request.setAttribute("msg", msg);
response.sendRedirect("/pro02");
//        response.setContentType("text/html;charset=UTF-8");
//        response.setCharacterEncoding("UTF-8");
//        view = request.getRequestDispatcher("/pro02");
//        view.forward(request, response);
    } else{
msg = "회원가입 실패";
response.sendRedirect("/pro02");
    }
    }
}

```

loginctrl.java

```

package shop.controller.custom;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
@WebServlet("/Login.do")
public class LoginCtrl extends HttpServlet {
@Override
    protected void service(HttpServletRequest request,
HttpServletRequest response) throws ServletException, IOException{
request.setAttribute("msg", "로그인 하시기 바랍니다.");
    RequestDispatcher view =
request.getRequestDispatcher("/custom/login.jsp");
view.forward(request, response);
    }
}

```


loginproctrl.java

```
package shop.controller.custom;
import shop.model.CustomDAO;
import shop.util.AES256;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/LoginPro.do")
public class LoginProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            String id = request.getParameter("id");
            String pw = request.getParameter("pw");
            String msg = "로그인 페이지";
            CustomDAO dao = new CustomDAO();
            boolean pass = dao.login(id, pw);
            HttpSession session = request.getSession();
            PrintWriter out = response.getWriter();
            RequestDispatcher view = null;
            if(pass) {
msg = "로그인 성공";
session.setAttribute("sid", id);
request.setAttribute("msg", msg);
response.sendRedirect("/pro02");
//            response.setContentType("text/html; charset=UTF-8");
//            response.setCharacterEncoding("UTF-8");
//            view = request.getRequestDispatcher("/pro02");
//            view.forward(request, response);
            } else{
response.sendRedirect("/pro02/Login.do");
            }
        }
    }
}
```

logoutctrl.java

```
package shop.controller.custom;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
@WebServlet("/Logout.do")
```

```

public class LogoutCtrl extends HttpServlet {
@Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        HttpSession session = request.getSession();
        session.invalidate();
        response.sendRedirect("/pro02");
    }
}

```

mypagectrl.java

```

package shop.controller.custom;
import shop.dto.Custom;
import shop.model.CustomDAO;
import shop.util.AES256;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
@WebServlet("/Mypage.do")
public class MypageCtrl extends HttpServlet {
@Override
        protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        HttpSession session = request.getSession();
        String id = (String) session.getAttribute("sid");
        CustomDAO dao = new CustomDAO();
        Custom cus = dao.getCustom(id);
        System.out.println(cus.toString());
        request.setAttribute("cus", cus);

        RequestDispatcher view =
        request.getRequestDispatcher("/custom/mypage.jsp");
        view.forward(request, response);
    }
}

```

mypageModifyctrl.java

```

package shop.controller.custom;
import shop.dto.Custom;
import shop.model.CustomDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

```

```

@WebServlet("/MypageModify.do")
public class MypageModifyCtrl extends HttpServlet {
    @Override
        protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        //      HttpSession session = request.getSession();
        String id = request.getParameter("sid");
        CustomDAO dao = new CustomDAO();
        Customcus = dao.getCustom(id);
        System.out.println(cus.toString());
        request.setAttribute("cus", cus);

        RequestDispatcher view =
        request.getRequestDispatcher("/custom/mypageModify.jsp");
        view.forward(request, response);
    }
}

```

mypageModifyProCtrl.java

```

package shop.controller.custom;
import shop.dto.Custom;
import shop.model.CustomDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
@WebServlet("/MypageModifyPro.do")
public class MypageModifyProCtrl extends HttpServlet {
    @Override
        protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        HttpSession session = request.getSession();
        String id = request.getParameter("id");
        String pw = request.getParameter("pw");
        String pw2 = request.getParameter("pw2");
        String birth = request.getParameter("birth");
        String tel = request.getParameter("tel");
        String email = request.getParameter("email");
        String address = request.getParameter("address1") + "<br>" +
        request.getParameter("address2") + "<br>" + request.getParameter("postcode");
        int cnt = 0;
        // 기존 비밀번호 확인을 위해 데이터를 가져옴
        CustomDAO dao = new CustomDAO();
        Customcus = dao.getCustom(id);
        if(cus.getPw().equals(pw)) {
        cus.setPw(pw2);
        cus.setBirth(birth);
        }
    }
}

```

```

cus.setTel(tel);
cus.setEmail(email);
cus.setAddress(address);
cnt = dao.modifyCustom(cus);
        if(cnt > 0) {
System.out.println("업데이트 완료");
response.sendRedirect("/pro02/Mypage.do");
}
} else{
System.out.println("비밀번호 불일치");
        RequestDispatcher view = request.getRequestDispatcher("/pro02");
view.forward(request, response);
}

}
}

```

2. 게시판 관련 컨트롤러

addnoticectrl.java

```

package shop.controller.notice;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/AddNotice.do")
public class AddNoticeCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            String msg = "관리자의 공지사항 글 등록폼이 로딩되었습니다.";
request.setAttribute("msg", msg);
                                RequestDispatcher view =
request.getRequestDispatcher("/WEB-INF/admin/addNotice.jsp");
view.forward(request, response);
}
}

```

addNoticeProCtrl.java

```

package shop.controller.notice;
import shop.dto.Notice;
import shop.model.NoticeDAO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

```

```

import java.io.PrintWriter;
@WebServlet("/AddNoticePro.do")
public class AddNoticeProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        //String title = request.getParameter("title");
        //String content = request.getParameter("content");
        Notice noti = new Notice();
        //noti.setTitle(title);
        noti.setTitle(request.getParameter("title"));
        noti.setContent(request.getParameter("content"));
        PrintWriter out = response.getWriter();
        NoticeDAO dao = new NoticeDAO();
        int a = dao.addNotice(noti);
        if(a>0) {
        response.sendRedirect("/pro02/NoticeList.do");
        } else{
        //request.sendRedirect("/AddNotice.do");
        out.println("<script>history.go(-1);</script>");
        }
        }
        }
}

```

DeleteNoticeCtrl.java

```

package shop.controller.notice;
import shop.model.NoticeDAO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/DeleteNotice.do")
public class DeleteNoticeCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        int no = Integer.parseInt(request.getParameter("no"));
        NoticeDAO dao = new NoticeDAO();
        int a = dao.deleteNotice(no);
        PrintWriter out = response.getWriter();
        if(a>0){
        response.sendRedirect("/pro02/NoticeList.do");
        } else{
        out.println("<script>history.go(-1);</script>");
        }
        }
        }
}

```

NoticeCtrl.java

```
package shop.controller.notice;
import shop.dto.Notice;
import shop.dto.Review;
import shop.model.NoticeDAO;
import shop.model.ReviewDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;
@WebServlet("/Notice.do")
public class NoticeCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            int no = Integer.parseInt(request.getParameter("no"));
            NoticeDAO dao = new NoticeDAO();
            Notice noti = dao.getNotice(no);
            request.setAttribute("noti", noti);

            RequestDispatcher view =
            request.getRequestDispatcher("/notice/getNotice.jsp");
            view.forward(request, response);
        }
    }
}
```

NoticeListCtrl.java

```
package shop.controller.notice;
import shop.dto.Notice;
import shop.model.NoticeDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;
@WebServlet("/NoticeList.do")
public class NoticeListCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            request.setAttribute("msg", "공지사항 목록을 출력합니다.");
            NoticeDAO dao = new NoticeDAO();
            List<Notice> notiList = dao.getNoticeList();
            request.setAttribute("notiList", notiList);

            RequestDispatcher view =
```

```

request.getRequestDispatcher("/notice/noticeList.jsp");
view.forward(request, response);
}
}

```

updateNoticeCtrl.java

```

package shop.controller.notice;
import shop.dto.Notice;
import shop.model.NoticeDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/UpdateNotice.do")
public class UpdateNoticeCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        String msg = "공지사항의 내용을 수정합니다.";
        int no = Integer.parseInt(request.getParameter("no"));
        NoticeDAO dao = new NoticeDAO();
        Notice noti = dao.getNotice(no);
        request.setAttribute("msg", msg);
        request.setAttribute("noti", noti);

        RequestDispatcher view =
        request.getRequestDispatcher("/WEB-INF/admin/updateNotice.jsp");
        view.forward(request, response);
    }
}

```

UpdateNoticeProCtrl.java

```

package shop.controller.notice;
import shop.dto.Notice;
import shop.model.NoticeDAO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/UpdateNoticePro.do")
public class UpdateNoticeProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        Notice noti = new Notice();
        noti.setNo(Integer.parseInt(request.getParameter("no")));

```



```

noti.setTitle(request.getParameter("title"));
noti.setContent(request.getParameter("content"));
    NoticeDAO dao = new NoticeDAO();
    int a = dao.updateNotice(noti);
    PrintWriter out = response.getWriter();
    if(a>0){
response.sendRedirect("/pro02/NoticeList.do");
    } else{
out.println("<script>history.go(-1);</script>");
    }
    }
}

```

3. 상품 관련 컨트롤러

addproductctrl.java

```

package shop.controller.product;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/AddProduct.do")
public class AddProductCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            String msg = "관리자의 공지사항 글 등록폼이 로딩되었습니다.";
            request.setAttribute("msg", msg);

            RequestDispatcher view =
            request.getRequestDispatcher("/WEB-INF/admin/addProduct.jsp");
            view.forward(request, response);
        }
    }
}

```

addproductproctrl.java

```

package shop.controller.product;
import com.oreilly.servlet.MultipartRequest;
import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;
import shop.dto.Product;
import shop.model.FileudDAO;
import shop.model.ProductDAO;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.File;

```



```

import java.io.IOException;
import java.util.Enumeration;
@WebServlet("/AddProductPro.do")
public class AddProductProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        String msg = "";
        ServletContext application = request.getServletContext();
        ProductDAO dao = new ProductDAO();
        try{
            String saveDirectory = application.getRealPath("/storage"); //실제
저장 경로
            int maxSize = 1024*1024*10; //10MB
            String encoding = "UTF-8";
            MultipartRequest mr = new MultipartRequest(request, saveDirectory,
maxSize, encoding, new DefaultFileRenamePolicy());
            Product pro = new Product();
            pro.setCate(mr.getParameter("cate"));
            pro.setCateno(mr.getParameter("cate"));
            pro.setPname(mr.getParameter("pname"));
            pro.setPcomment(mr.getParameter("pcomment"));
            pro.setPlist(mr.getParameter("plist"));
            pro.setPrice(Integer.parseInt(mr.getParameter("price")));
            pro.setImgSrc1(mr.getParameter("imgsrc1"));
            pro.setImgSrc2(mr.getParameter("imgsrc2"));
            pro.setImgSrc3(mr.getParameter("imgsrc3"));
            File upfile = null;
            Enumeration files = mr.getFileNames();
            int idx = 1;
            String item;
            String oriFile = ""; //실제 첨부된 파일경로와 이름
            String fileName = ""; //파일 이름만 추출
            System.out.println("파일추출");
            while(files.hasMoreElements()) {
            item = (String) files.nextElement();
            oriFile = mr.getOriginalFileName(item); //실제 첨부된 파일경로와 이름
            fileName = mr.getFilesystemName(item); //파일 이름만 추출
                if(fileName!=null) {
            upfile = mr.getFile(item); //실제 업로드
                if(upfile.exists()) {
                    long filesize = upfile.length();
                    if(idx==1) {
            pro.setImgSrc1(upfile.getName());
            System.out.println("img1 in");
            } else if(idx==2){
            pro.setImgSrc2(upfile.getName());
            System.out.println("img2 in");
            } else if(idx==3){
            pro.setImgSrc3(upfile.getName());

```

```

System.out.println("img3 in");
}
msg = "파일 업로드 성공";
} else{
msg = "파일 업로드 실패";
}
}
idx++;
}

/*
pro.setImgSrc1(upfile.get());
file.setFilename(upfile.getName());*/
int cnt = dao.addProduct(pro);
if(cnt > 0) {
System.out.println("db 업로드 완료");
}
} catch(Exception e){
System.out.println(e.getMessage());
}
response.sendRedirect("/pro02/proList.do");
}
}

```

addReceivectl.java

```

package shop.controller.product;
import shop.dto.Product;
import shop.model.ProductDAO;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
import java.util.List;
@WebServlet("/AddReceive.do")
public class AddReceiveCtrl extends HttpServlet {
@Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        String msg = "관리자의 상품 입고 품이 로딩되었습니다.";
        request.setAttribute("msg", msg);
        ProductDAO dao = new ProductDAO();
        List<Product> proList = dao.getProductList();
        request.setAttribute("proList", proList);

        RequestDispatcher view =
        request.getRequestDispatcher("/WEB-INF/admin/addReceive.jsp");
        view.forward(request, response);
    }
}

```

addreceivewproctrl.java

```

package shop.controller.product;

```

```

import shop.dto.Receive;
import shop.model.ProductDAO;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
@WebServlet("/AddReceivePro.do")
public class AddReceiveProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            ServletContext application = request.getServletContext();
            String home = application.getContextPath();
            Receiver rec = new Receive();
            rec.setPno(Integer.parseInt(request.getParameter("no")));
            rec.setAmount(Integer.parseInt(request.getParameter("amount")));
            rec.setRprice(Integer.parseInt(request.getParameter("price")));
            ProductDAO dao = new ProductDAO();
            int cnt = dao.addReceive(rec);
            if(cnt > 0){
                response.sendRedirect(home + "/ProList.do");
            } else{
                response.sendRedirect(home + "/AddReceive.do");
            }
        }
    }
}

```

delproductctrl.java

```

package shop.controller.product;
import shop.model.ProductDAO;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
@WebServlet("/ProductDel.do")
public class DelProductCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            int no = Integer.parseInt(request.getParameter("no"));
            System.out.println(no);
            ProductDAO dao = new ProductDAO();
            int cnt = dao.delProduct(no);
            if(cnt > 0) {
                System.out.println("product 삭제 성공, no="+no);
            }

            RequestDispatcher view =
            request.getRequestDispatcher("/pro02/ProList.do");
            view.forward(request, response);
        }
    }
}

```

```
}
```

```
productctrl.java
```

```
package shop.controller.product;
import shop.dto.Payment;
import shop.dto.Product;
import shop.dto.Review;
import shop.model.DeliveryDAO;
import shop.model.PaymentDAO;
import shop.model.ProductDAO;
import shop.model.ReviewDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
@WebServlet("/Product.do")
public class ProductCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            int no = Integer.parseInt(request.getParameter("no"));
            HttpSession session = request.getSession();
            String sid = (String) session.getAttribute("sid");
            // 상품 상세 내용 정보 추출
            ProductDAO dao = new ProductDAO();
            Product pro = dao.getProduct(no);
            System.out.println(pro.toString());
            request.setAttribute("pro", pro);

            // 리뷰 목록 추출
            ReviewDAO dao2 = new ReviewDAO();
            List<Review> revList = dao2.getReviewList(no);
            // System.out.println(revList.toString());
            for(Review r: revList) {
                r.toString();
            }
            System.out.println("revList 보내기");
            request.setAttribute("revList", revList);
            // 해당 상품 구매자 확인, 구매자 인 경우 리뷰 작성 가능
            PaymentDAO dao3 = new PaymentDAO();
            int check1 = dao3.getPaymentBuyCustom(no, sid);
            DeliveryDAO dao4 = new DeliveryDAO();
            int check2 = dao4.getDelivery(no, sid);
            int check = check1 + check2;
```

```
request.setAttribute("check", check);
```

```
request.getRequestDispatcher(view  
request.getRequestDispatcher("/product/getProduct.jsp");  
view.forward(request, response);  
}  
}
```

prolistctrl.java

```
package shop.controller.product;  
import shop.dto.Product;  
import shop.dto.Review;  
import shop.model.ProductDAO;  
import shop.model.ReviewDAO;  
import javax.servlet.RequestDispatcher;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
@WebServlet("/ProList.do")  
public class ProListCtrl extends HttpServlet {  
    @Override  
        protected void doGet(HttpServletRequest request,  
        HttpServletResponse response) throws ServletException, IOException {  
            ProductDAO dao = new ProductDAO();  
            List<Product> proList = new ArrayList<>();  
            String cate = request.getParameter("cate");  
            System.out.println(cate);  
            if(cate == null) {  
                proList = dao.getProductList();  
                System.out.println(proList.toString());  
                request.setAttribute("proList", proList);  
            } else if(cate.equals("초등")) {  
                proList = dao.getCateProductList("초등");  
                System.out.println(proList.toString());  
                request.setAttribute("proList", proList);  
            } else if(cate.equals("중등")) {  
                proList = dao.getCateProductList("중등");  
                System.out.println(proList.toString());  
                request.setAttribute("proList", proList);  
            } else if(cate.equals("고등")) {  
                proList = dao.getCateProductList("고등");  
                System.out.println(proList.toString());  
                request.setAttribute("proList", proList);  
            }  
            System.out.println("상품목록 조회" + cate);  
        }  
    }  
}
```

```

RequestDispatcher view =
request.getRequestDispatcher("/product/proList.jsp");
view.forward(request, response);
}
}

```

updateproductctrl.java

```

package shop.controller.product;

import shop.dto.Category;
import shop.dto.Product;
import shop.model.ProductDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;
@WebServlet("/UpdateProduct.do")
public class UpdateProductCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            String msg = "상품 정보를 수정합니다.";
            int no = Integer.parseInt(request.getParameter("no"));
            ProductDAO dao = new ProductDAO();
            Product pro = dao.getProduct(no);
            List<Category> cateList = dao.getCategoryList();
request.setAttribute("cateList", cateList);
request.setAttribute("msg", msg);
request.setAttribute("pro", pro);

RequestDispatcher view =
request.getRequestDispatcher("/product/updateProduct.jsp");
view.forward(request, response);
}
}

```

updateproductproctrl.java

```

package shop.controller.product;

import com.oreilly.servlet.MultipartRequest;
import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;
import shop.dto.Product;
import shop.model.ProductDAO;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;

```

```

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
@WebServlet("/UpdateProductPro.do")
public class UpdateProductProCtrl extends HttpServlet {
    protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        String msg = "";
        ServletContext application = request.getServletContext();
        String home = application.getContextPath();
        try{
            String saveDirectory = application.getRealPath("/storage"); //실제
저장 경로
            int maxSize = 1024*1024*10; //10MB
            String encoding = "UTF-8";
            MultipartRequest mr = new MultipartRequest(request, saveDirectory,
maxSize, encoding, new DefaultFileRenamePolicy());
            Product pro = new Product();
            pro.setNo(Integer.parseInt(mr.getParameter("no")));
            pro.setPname(mr.getParameter("pname"));
            pro.setPcomment(mr.getParameter("pcomment"));
            pro.setPlist(mr.getParameter("plist"));
            pro.setPrice(Integer.parseInt(mr.getParameter("price")));
            pro.setCate(mr.getParameter("cate"));
            File upfile = null;
            Enumeration files = mr.getFileNames();
            int idx = 1;
            String item;
            String oriFile = ""; //실제 첨부된 파일경로와 이름
            String fileName = ""; //파일 이름만 추출
            System.out.println("파일 추출");
            while(files.hasMoreElements()) {
                item = (String) files.nextElement();
                oriFile = mr.getOriginalFileName(item); //실제 첨부된 파일경로와 이름
                fileName = mr.getFilesystemName(item); //파일 이름만 추출
                if(fileName!=null) {
                    upfile = mr.getFile(item); //실제 업로드
                    if(upfile.exists()) {
                        long fileSize = upfile.length();
                        if(idx==1) {
                            pro.setImgSrc1(upfile.getName());
                            System.out.println("img1 in");
                        } else if(idx==2){
                            pro.setImgSrc2(upfile.getName());
                            System.out.println("img2 in");
                        }
                    }
                }
                idx++;
            }
        } catch (Exception e) {
            msg = e.getMessage();
        }
    }
}

```

```

} else if(idx==3){
pro.setImgSrc3(upfile.getName());
System.out.println("img3 in");
}
msg = "파일 업로드 성공";
} else{
msg = "파일 업로드 실패";
}
}
idx++;
}
System.out.println(fileName);
        ProductDAOdao = new ProductDAO();
        int cnt = 0;
cnt = dao.updateProduct(pro);
        if(cnt>0){
response.setContentType("text/html; charset=UTF-8");
        PrintWriterout = response.getWriter();
out.println("<script>alert('상품 수정이 완료되었습니다.');

```

updateReceiveCtrl.java

```

package shop.controller.product;
import shop.dto.Product;
import shop.model.ProductDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;

```



```

@WebServlet("/UpdateReceive.do")
public class UpdateReceiveCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        String msg = "상품 업데이트 페이지 실행";
        request.setAttribute("msg", msg);
        ProductDAO dao = new ProductDAO();
        List<Product> proList = dao.getProductList();
        request.setAttribute("proList", proList);

        RequestDispatcher view =
        request.getRequestDispatcher("/WEB-INF/admin/updateReceive.jsp");
        view.forward(request, response);
    }
}

```

updateReceiveProCtrl.java

```

package shop.controller.product;
import shop.dto.Receive;
import shop.model.ProductDAO;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/UpdateReceivePro.do")
public class UpdateReceiveProCtrl extends HttpServlet {
    @Override
        protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("utf-8");
        try{
            int pno = Integer.parseInt(request.getParameter("pno"));
            int amount = Integer.parseInt(request.getParameter("amount"));
            String msg = "";
            ServletContext application = request.getServletContext();
            ProductDAO dao = new ProductDAO();
            Receive rec = new Receive();
            rec.setPno(pno);
            rec.setAmount(amount);
            int cnt = 0;
            cnt = dao.updateReceive(rec);
            if(cnt > 0) {
                System.out.println("db 업로드 완료");
                response.setContentType("text/html; charset=UTF-8");
                PrintWriter out = response.getWriter();
                out.println("<script>alert('상품추가 등록이 완료되었습니다.');

```

```

out.println("<script> location.href= '/pro02/ProList.do'; </script>");
out.flush();
}else{
response.setContentType("text/html; charset=UTF-8");
PrintWriterout = response.getWriter();
out.println("<script>alert('상품추가          등록에          실패하였습니다');
location.href='/pro02/UpdateReceive.do';</script>");
out.flush();

//response.sendRedirect(home+"/AddProduct.do");
}
} catch(Exceptione){
System.out.println(e.getMessage());
}
}
}
}

```

4. 결제 배송에 관한 모든 컨트롤러

cusDelivery.java

```

package shop.controller.delivery;
//import shop.dto.DeliveryInfoView;
import shop.dto.DeliveryInfoView;
import shop.model.DeliveryDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
// 배송 중
@WebServlet("/CusDelivery.do")
public class CusDelivery extends HttpServlet {
@Override
protected void service(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException{
    int sno = Integer.parseInt(request.getParameter("sno"));
    DeliveryDAO dao = new DeliveryDAO();
    DeliveryInfoView div1 = new DeliveryInfoView();
    div1 = dao.deliveryInfoViewList(sno);
    request.setAttribute("deli", div1);

    RequestDispatcher view =
request.getRequestDispatcher("/delivery/cusDelivery.jsp");
view.forward(request, response);
}
}

```

DeliveryListCtrl.java

```

package shop.controller.delivery;
import shop.dto.Delivery;
import shop.model.DeliveryDAO;
import javax.servlet.RequestDispatcher;

```

```

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
// 배송 시작
@WebServlet("/AdminDeliveryList.do")
public class DeliveryListCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            String msg = "배송 시작 하기";
            List<Delivery> deliveryList = new ArrayList<>();
            DeliveryDAO dao = new DeliveryDAO();
            int pstate = Integer.parseInt(request.getParameter("pstate"));
            deliveryList = dao.getDeliveryList(pstate);

            String title = "배송관리";

            request.setAttribute("deliveryList", deliveryList);
            if(pstate == 0) {
                "출고처리";
            } else if(pstate == 1) {
                "배송중";
            } else if(pstate == 2) {
                "배송완료";
            } else{
                "구매완료";
            }
            System.out.println(title);
            request.setAttribute("title", title);
            request.setAttribute("msg", msg);

            RequestDispatcher view =
            request.getRequestDispatcher("/WEB-INF/admin/DeliveryList.jsp");
            view.forward(request, response);
        }
    }

```

UpdDeliveryCtrl.java

```

package shop.controller.delivery;
import shop.dto.Delivery;
import shop.model.DeliveryDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/UpdateDeliveryPost.do")
public class UpdDeliveryCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            Delivery deli = new Delivery();
            DeliveryDAO dao = new DeliveryDAO();
            int dno = Integer.parseInt(request.getParameter("dno"));
            deli = dao.getDelivery(dno);
            request.setAttribute("deli", deli);

            RequestDispatcher view =
            request.getRequestDispatcher("/WEB-INF/admin/updateDelivery.jsp");
            view.forward(request, response);
        }
    }

```

UpdDeliveryProCtrl.java

```

package shop.controller.delivery;
import shop.dto.Delivery;
import shop.model.DeliveryDAO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/UpdateDeliveryPostPro.do")
public class UpdDeliveryProCtrl extends HttpServlet {
    @Override
        protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            int pstate = Integer.parseInt(request.getParameter("pstate"));
            int dno = Integer.parseInt(request.getParameter("dno"));
            String pcom = request.getParameter("pcom");
            String ptel = request.getParameter("ptel");
            String bcode = request.getParameter("bcode");
            String rdate = request.getParameter("rdate");
            Delivery deli = new Delivery();
            deli.setPcom(pcom);
            deli.setPtel(ptel);
            deli.setPstate(pstate);
            deli.setBcode(bcode);
            deli.setDno(dno);
            deli.setRdate(rdate);
            DeliveryDAO dao = new DeliveryDAO();
            int cnt = 0;
            cnt = dao.updateDelivery(deli);
            if(cnt > 0) {

```

```

System.out.println("배송 정보 수정 완료");
} else{
System.out.println("배송 정보 수정 실패");
}
response.sendRedirect("/pro02/AdminDeliveryList.do?pstate="+pstate);
// RequestDispatcher view =
request.getRequestDispatcher("/pro02/AdminDeliveryList.do?pstate"+pstate);
// view.forward(request, response);
}
}

```

AddPaymentCtrl.java

```

package shop.controller.payment;
import shop.dto.Custom;
import shop.dto.Product;
import shop.model.CustomDAO;
import shop.model.ProductDAO;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/AddPayment.do")
public class AddPaymentCtrl extends HttpServlet {
    @Override
        protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");
        int pno = Integer.parseInt(request.getParameter("pno"));
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();
        String cid = (String) session.getAttribute("sid");
        String from;
        String cartno;
        if(request.getParameter("from") != null) {
        from = request.getParameter("from");
        request.setAttribute("from", from);
        } else{
        }
        if(request.getParameter("cartno") != null) {
        cartno = request.getParameter("cartno");
        request.setAttribute("cartno", cartno);
        } else{
        }
    }
}

```

```

        ProductDAOdao = newProductDAO();
        Productpro = dao.getProduct(pno);
        intamount = dao.getAmount(pno);
        CustomDAOcusDao = newCustomDAO();
        Customcus = cusDao.getCustom(cid);
request.setAttribute("pro", pro);
request.setAttribute("amount", amount);
request.setAttribute("cus", cus);
System.out.println("amount: "+ amount);
        if(pro!=null) {
                                                    RequestDispatcherview =
request.getRequestDispatcher("/payment/addPayment.jsp");
view.forward(request, response);
} else{
        //request.sendRedirect("/AddNotice.do");
out.println("<script>history.go(-1);</script>");
}
}
}
}

```

AddPaymentProCtrl.java

```

package shop.controller.payment;
import shop.dto.Delivery;
import shop.dto.Payment;
import shop.dto.Serve;
import shop.model.CartDAO;
import shop.model.DeliveryDAO;
import shop.model.PaymentDAO;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/AddPaymentPro.do")
public class AddPaymentProCtrl extends HttpServlet {
@Override
        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException{
        //결제 처리(PaymentDAO.addPayment(pay))
        Paymentpay = newPayment();
pay.setCid(request.getParameter("cid"));
pay.setPno(Integer.parseInt(request.getParameter("pno")));
pay.setAmount(Integer.parseInt(request.getParameter("amount")));
pay.setPmethod(request.getParameter("pmethod"));
pay.setPcom(request.getParameter("pcom2"));
pay.setCnum(request.getParameter("cnum"));
pay.setPayprice(Integer.parseInt(request.getParameter("payAmount")));
        PaymentDAOpayDAO = newPaymentDAO();
        intcnt1 = payDAO.addPayment(pay);
        //출고 처리(PaymentDAO.addServe(serv))

```

```

        Servserv = newServe();
serv.setPno(Integer.parseInt(request.getParameter("pno")));
serv.setAmount(Integer.parseInt(request.getParameter("amount")));
        intsprice = (int) Double.parseDouble(request.getParameter("sprice"));
serv.setSprice(sprice);
        intcnt2 = payDAO.addServe(serv);
        //배송 등록(DeliveryDAO.addDelivery(del))
        Deliverydel = newDelivery();
del.setSno(payDAO.getSno());
del.setCid(request.getParameter("cid"));
del.setDaddr(request.getParameter("address1")+"<br>" + request.getParameter("address2")+"<br>" + request.getParameter("postcode"));
del.setCustel(request.getParameter("custel"));
        DeliveryDAOdeliDAO = newDeliveryDAO();
        intcnt3 = deliDAO.addDelivery(del);
        //장바구니에서 결제한 정보라면 (CartDAO.delCart(cartno));
        CartDAOcartDAO = newCartDAO();
        Stringfrom = request.getParameter("from");
        intcartno = 0;
        intcnt4 = 0;
        if(from.equals("cart")){
cartno = Integer.parseInt(request.getParameter("cartno"));
cnt4 = cartDAO.delCart(cartno);
}

        intpno = Integer.parseInt(request.getParameter("pno"));
        if(cnt1 > 0&& cnt2 > 0&& cnt3 > 0){
response.sendRedirect(request.getContextPath()+"/ProList.do");
} else{
response.sendRedirect(request.getContextPath()+"/AddPayment.do?pno="+pno);
}
}
}
}

```

BuyPaymentCtrl.java

```

package shop.controller.payment;
import shop.model.DeliveryDAO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/BuyPayment.do")
public class BuyPaymentCtrl extends HttpServlet {
@Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
        int sno = Integer.parseInt(request.getParameter("sno"));
        DeliveryDAO deliDAO = new DeliveryDAO();

```



```

        boolean ck = deliDAO.buyDelivery(sno);
        if(ck) {
            System.out.println("구매 확정");
            response.setContentType("text/html; charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("<script>alert('구매가          확정되었습니다.');"
            location.href='/pro02/PaymentList.do';</script>");
            out.flush();
        } else{
            response.setContentType("text/html; charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("<script>alert('오류가          발생했습니다.');"
            location.href='/pro02/PaymentList.do';</script>");
            out.flush();
        }
    }
}

```

PaymentListCtrl.java

```

package shop.controller.payment;
import shop.dto.PaymentVO;
import shop.model.PaymentDAO;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
import java.util.List;
@WebServlet("/PaymentList.do")
public class PaymentListCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            HttpSession session = request.getSession();
            String cid = (String) session.getAttribute("sid");
            PaymentDAO payDAO = new PaymentDAO();
            List<PaymentVO> payList = payDAO.getCidPaymentList(cid);

            request.setAttribute("payList", payList);

            RequestDispatcher view =
            request.getRequestDispatcher("/payment/paymentList.jsp");
            view.forward(request, response);
        }
    }
}

```

ReturnPaymentCtrl.java

```

package shop.controller.payment;
import shop.dto.Payment;
import shop.model.PaymentDAO;
import javax.servlet.*;

```



```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.io.IOException;
@WebServlet("/ReturnPayment.do")
public class ReturnPaymentCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            int sno = Integer.parseInt(request.getParameter("sno"));
            PaymentDAO payDAO = new PaymentDAO();
            Payment pay = payDAO.getPayment(sno);
            int cnt = payDAO.returnPayment(sno, pay.getPno(), pay.getAmount(),
            pay.getCid());
            response.sendRedirect(request.getContextPath()+"/PaymentList.do");
        }
    }
}
```

해당 능력단위 요소	화면 구현하기	(20점)
수행4. 작성 내용		

1. 회원 관련 부

회원 가입

 **회원가입**

아이디

아이디 (6~16자)

중복확인

아직 아이디 중복 체크를 하지 않으셨습니다.

비밀번호

비밀번호 (문자, 숫자, 특수문자 포함 8~20자)

비밀번호 확인

비밀번호 확인

이름

이름

생년월일

연도-월-일

전화번호

전화번호

이메일 주소

이메일

약관

제 1 장 총칙

제 1 조 (목적) 이 이용약관은 "단체표준종합정보센터 (이하 "당 사이트")"에서 제공하는 인터넷 서비스(이하 "서비스")의 가입조건, 당 사이트와 이용자의 권리, 의무, 책임사항과 기타 필요한 사항을 규정함을 목적으로 합니다.

☐ 약관의 동의

기본 주소 입력

상세 주소 입력

우편번호

우편번호 검색

가입 완료

가입 취소

```
join.jsp
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <!-- /web/member/login.jsp -->
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>회원 정보수정</title>
<%@ include file="../head.jsp" %>
  <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
  <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
  <!-- 필요한 폰트를 로딩 : 구글 웹 폰트에서 폰트를 선택하여 해당 내용을 붙여 넣
기 -->
  <link rel="stylesheet"href="${path}/css/font.css">
  <!-- 필요한 플러그인 연결 -->
  <script src="https://code.jquery.com/jquery-latest.js"></script>
  <link rel="stylesheet"href="${path}/css/common.css">
  <link rel="stylesheet"href="${path}/css/hd.css">

  <style>
    /* 본문 영역 스타일 */
    .contents{
      clear: both;
      min-height: 250vh;
      background-image: url("${path}/img/login.jpg");
      background-repeat: no-repeat;
      background-position: center -250px;
    }

    .contents::after{
      content: "";
      clear: both;
      display: block;
      width: 100%;
    }

    .page{
      clear: both;
      width: 100vw;
      height: 150vh;
      position: relative;
    }

    .page::after{
      content: "";
      display: block;
      width: 100%;
      clear: both;
    }

    .page_wrap{
      clear: both;
      width: 1200px;

```

```

        height: auto;
        margin: 0 auto;
    }

    .page_tit{
        font-size: 48px;
        text-align: center;
        padding-top: 1em;
        color: #fff;
        padding-bottom: 2.4rem;
    }

    .breadcrumb{
        clear: both;
        width: 1200px;
        margin: 0 auto;
        text-align: right;
        color: #fff;
        padding-top: 28px;
    }

    .breadcrumb a{
        color: #fff;
    }

d {

        box-sizing: border-box;
        font-family: 'Noto Sans KR', sans-serif;
    }

    .join_form{
        margin: 0 auto;
        margin-top: 50px;
        background: #eaf4fa;
        border-radius: 5px;
        width: 600px;
        height: auto;
        max-width: 100%;
        overflow: hidden;
        box-shadow: 0 0 25px rgba(0, 0, 0, 0.2);
    }

    .header{
        padding: 40px 40px;
        text-align: center;
    }

    .header img{
        display: inline-block;
    }

    .header h2{
        display: inline-block;
        margin: 0;
        font-size: 30px;
        font-weight: bolder;
    }

```

```

        padding-bottom: 15px;
        padding-left: 5px;
    }

    .frm1{
        padding: 30px 40px;
    }
    forminf {
        padding-bottom: 10px;
    }

    .register_info_id{
        position: relative;
    }

    .register_info_id button{
        position: absolute;
        width: 90px;
        height: 40px;
        top: 0;
        bottom: 0;
        right: 5px;
        margin: 32px 0;
        border-radius: 3px;
        background: cornflowerblue;
        color: #fff;
        border: 0;
    }

    .register_info input{
        border: 1.5px solid #535053;
        border-radius: 3px;
        display: block;
        font-size: 11px;
        padding: 5px;
        width: 80%;
    }

    .submit{
        background-color: cornflowerblue;
        border: 2px solid cornflowerblue;
        width: 120px;
        border-radius: 5px;
        padding: 5px;
        color: #ffffff;
    }

```

```

        float: left;
    }

    .agree_fr{
        margin: 20px auto;
        border: 1px solid #eee;
        padding: 20px;
        overflow-y: auto;
        height: 100px;
        white-space: pre-wrap;
    }

    .agree_fr2{
        border-radius: 5px;
        height: 100px;
        border: 1px solid #d9d6d6;
        background-color: white;
        white-space: pre-line;
        overflow-y: scroll;
        margin: 0 auto 20px;
        padding: 10px;
    }

    .frm1 p{
        text-align: right;
    }

    .frm1 input{
        display: inline-block;
        width: 100%;
        border: 1px solid #d9d6d6;
        padding: 17px 25px;
        box-sizing: border-box;
        margin: 0;
        border-radius: 5px;
    }

    .register_info{
        padding: 10px 0;
    }

    input::placeholder{
        font-size: 15px;
        color: #9d9d9d;
    }

    label{
        padding-bottom: 10px;
        display: block;
        font-weight: bold;
        font-size: 16px;
    }

```

```

        color: #000;
    }

    label > span{
        font-weight: normal;
        color: red;
        font-size: 13px;
        padding-left: 10px;
    }

    .btn_fr input{
        display: inline-block;
        width: auto;
        float: left;
    }

    .btn_fr label{
        padding-left: 25px;
    }

    input.inbtn{
    }

    button.submit{
        display: block;
        height: 50px;
        background: cornflowerblue;
        border: 0;
        border-radius: 5px;
        width: 50%;
        color: #fff;
        font-size: 18px;
        float: left;
        margin: 0 5px;
        padding: 0;
        box-shadow: 2px 3px 7px 0px #0000005e;
    }

    button.reset{
        height: 50px;
        background: gray;
        border: 0;
        border-radius: 5px;
        width: 50%;
        color: #fff;
        font-size: 18px;
        padding: 0;
        margin: 0 5px;
        box-shadow: 2px 3px 7px 0px #0000005e;
    }

    .btn1{
        color: #fff;
        width: 70%;
        display: flex;
        margin: 20px auto;
    }

```

```

</style>
<link rel="stylesheet"href="${path}/css/ft.css">
</head>
<body>
<div class="wrap">
  <header class="hd"id="hd">
<%@ include file="../../../header.jsp" %>
  </header>
  <div class="contents"id="contents">
    <div class="breadcrumb">
      <p><a href="">HOME</a>&gt; <span>회원가입</span></p>
    </div>
    <section class="page"id="page1">
      <div class="page_wrap">
        <h2 class="page_tit">회원 가입</h2>

        <div class="join_form">
          <div class="header">
            
            <h2>회원가입</h2>
          </div>

          <form class="frm1"action="${path
}/JoinPro.do"method="post"onsubmit="return joinCheck(this)">
            <div class="register_info_id">
              <label>아이디<!--<span>사용할 수 없는 아이디 입니다
</span>--%></label>

              <input
type="text"name="id"id="id"class="indata"placeholder="아이디
(6~16
자)"pattern="^[a-z0-9]{6,16}"maxlength="16"autofocus required/>

              <button
type="button"id="ck_btn"class="btn-primary"onclick="idCheck()">중복확인
</button>
<!-- <input type="button"id="ck_btn" class="btn btn-primary"value="중복 확인
"onclick="idCheck()">--%>

              <input
type="hidden"name="idck"id="idck"value="no">
            </div>
            <div>
              <c:if test="${empty qid }">
                <p id="msg"style="padding-left:0.5rem">아직 아
이디 중복 체크를 하지 않으셨습니다.</p>
              </c:if>
              <c:if test="${not empty qid }">
                <p id="msg"style="padding-left:0.5rem">아이디
중복 체크후 수정하였습니다.</p>
              </c:if>
            </div>
            <div class="register_info">
              <label>비밀번호<!--<span>20자 이내 비밀번호를 입력하

```



```

세요</span>--%></label>
<input
type="password"name="pw"id="pw"class="indata"
placeholder="비밀번호 (문자, 숫자, 특수문자 포
함 8~20자)"required="">
</div>
<div class="register_info">
<label>비밀번호 확인<%--<span>비밀번호가 일치하지 않
습니다</span>--%></label>
<input
type="password"name="pw2"id="pw2"class="indata"placeholder="비밀번호
"required="">
</div>
<div class="register_info">
<label>이름</label>
<input
type="text"name="name"id="name"class="indata"placeholder="이름"required="">
</div>
<div class="register_info">
<label>생년월일</label>
<td><input
type="date"name="birth"id="birth"placeholder="생년월일
"class="form-control"required></td>
</div>
<div class="register_info">
<label>전화번호</label>
<input
type="tel"name="tel"id="tel"class="indata"placeholder="전화번호"required="">
</div>
<div class="register_info">
<label>이메일 주소</label>
<input
type="email"name="email"id="email"class="indata"placeholder="이 메 일
"required="">
</div>
<div class="register_info">
<label>약관</label>
<div class="agree_fr2">

```

제 1 장 총칙

제 1 조 (목적) 이 이용약관은 “단체표준종합정보센터 (이하 "당 사이트")”에서 제공하는 인터넷 서비스(이하 '서비스')의 가입조건, 당 사이트와 이용자의 권리, 의무, 책임사항과 기타 필요한 사항을 규정함을 목적으로 합니다.

제 2 조 (용어의 정의)

1. "이용자"라 함은 당 사이트에 접속하여 이 약관에 따라 당 사이트가 제공하는 서비스를 받는 회원 및 비회원을 말합니다.
2. “단체표준종합정보센터”에서 제공하는 단체표준, 단체인증, 예고현황 정보를 말합니다.
3. "회원"이라 함은 서비스를 이용하기 위하여 당 사이트에 개인정보를 제공하여 아이디 (ID)와 비밀번호를 부여 받은 자를 말합니다.
4. “비회원”이하 함은 회원으로 가입하지 않고 " 단체표준종합정보센터"에서 제공하는 서

비스를 이용하는 자를 말합니다.

5. "회원 아이디(ID)"라 함은 회원의 식별 및 서비스 이용을 위하여 자신이 선정한 문자 및 숫자의 조합을 말합니다.

6. "비밀번호"라 함은 회원이 자신의 개인정보 및 직접 작성한 비공개 콘텐츠의 보호를 위하여 선정한 문자, 숫자 및 특수문자의 조합을 말합니다.

제 3 조 (이용약관의 효력 및 변경)

1. 당 사이트는 이 약관의 내용을 회원이 알 수 있도록 당 사이트의 초기 서비스화면에 게시합니다. 다만, 약관의 내용은 이용자가 연결화면을 통하여 볼 수 있도록 할 수 있습니다.

2. 당 사이트는 이 약관을 개정할 경우에 적용일자 및 개정사유를 명시하여 현행 약관과 함께 당 사이트의 초기화면 또는 초기화면과의 연결화면에 그 적용일자

7일 이전부터 적용일자 전일까지 공지합니다. 다만, 회원에게 불리하게 약관내용을 변경하는 경우에는 최소한 30일 이상의 사전 유예기간을 두고 공지합니다. 이

경우 당 사이트는 개정 전 내용과 개정 후 내용을 명확하게 비교하여 이용자가 알기 쉽도록 표시합니다.

3. 당 사이트가 전항에 따라 개정약관을 공지하면서 “개정일자 적용 이전까지 회원이 명시적으로 거부의 의사표시를 하지 않는 경우 회원이 개정약관에 동의한

것으로 봅니다.”라는 취지를 명확하게 공지하였음에도 회원이 명시적으로 거부의 의사표시를 하지 않은 경우에는 개정약관에 동의한 것으로 봅니다. 회원이 개정약관에

동의하지 않는 경우 당 사이트 이용계약을 해지할 수 있습니다.

제 4 조 (약관 외 준칙)

1. 이 약관은 당 사이트가 제공하는 서비스에 관한 이용안내와 함께 적용됩니다.

2. 이 약관에 명시되지 아니한 사항은 관계법령의 규정이 적용됩니다.

제 2 장 이용계약의 체결

제 5 조 (이용계약의 성립 등)

이용계약은 이용고객이 당 사이트가 정한 약관에 「동의합니다」를 선택하고, 당 사이트가 정한 회원가입양식을 작성하여 서비스 이용을 신청한 후, 당 사이트가 이를

승낙함으로써 성립합니다.

제 6 조 (회원가입)

서비스를 이용하고자 하는 고객은 당 사이트에서 정한 회원가입양식에 개인정보를 기재하여 가입을 하여야 합니다.

제 7 조 (개인정보의 보호 및 사용)

당 사이트는 관계법령이 정하는 바에 따라 회원 등록정보를 포함한 회원의 개인정보를 보호하기 위해 노력합니다. 회원 개인정보의 보호 및 사용에 대해서는 관련법령

및 당 사이트의 개인정보 보호정책이 적용됩니다. 다만, 당 사이트 이외에 링크된 사이트에서는 당 사이트의 개인정보 보호정책이 적용되지 않습니다.

제 8 조 (이용 신청의 승낙과 제한)

1. 당 사이트는 제6조의 규정에 의한 이용신청고객에 대하여 약관에 정하는 바에 따라 서비스 이용을 승낙합니다.

2. 당 사이트는 아래사항에 해당하는 경우에 대해서 회원가입을 승낙하지 아니하거나 이후 사전 통보 없이 취소할 수 있습니다.

- 회원가입 신청 시 내용을 허위로 기재한 경우
- 기타 규정한 제반사항을 위반하며 신청하는 경우
- 다른 사람의 당 사이트 이용을 방해하거나 그 정보를 도용하는 등의 행위를 하였을 경우
- 당 사이트를 이용하여 법령과 본 약관이 금지하는 행위를 하는 경우

1. 당 사이트는 이용고객에 대하여 약관에 정하는 바에 따라 자신이 선정한 회원 아이디를 부여합니다.

2. 회원 아이디는 원칙적으로 변경이 불가하며 부득이한 사유로 인하여 변경 하고자 하는 경우에는 해당 아이디를 해지하고 재가입해야 합니다.

3. 회원은 회원가입 시 기재한 개인정보가 변경되었을 경우 온라인으로 직접 수정할 수 있

습니다. 이때 변경하지 않은 정보로 인해 발생하는 문제에 대한 책임은 회원에게 있습니다.

제 3 장 계약 당사자의 의무

제 10 조 ("단체표준종합정보센터"의 의무)

1. 당 사이트는 이용고객이 희망한 서비스 제공 개시일에 특별한 사정이 없는 한 서비스를 이용할 수 있도록 하여야 합니다.
2. 당 사이트는 개인정보 보호를 위해 보안시스템을 구축하며 개인정보 보호정책을 공시하고 준수합니다.
3. 당 사이트는 회원으로부터 제기되는 의견이 합당하다고 판단될 경우에는, 적절한 조치를 취하여야 합니다.
4. 당 사이트는 전시, 사변, 천재지변, 비상사태, 현재의 기술로는 해결이 불가능한 기술적 결함 기타 불가항력적 사유 및 이용자의 귀책사유로 인하여 발생한 이용자의 손해, 손실, 기타 모든 불이익에 대하여 어떠한 책임도 지지 않습니다.

제 11 조 (회원의 의무)

1. 이용자는 회원가입 신청 또는 회원정보 변경 시 실명으로 모든 사항을 사실에 근거하여 작성하여야 하며, 허위 또는 타인의 정보를 등록할 경우 일체의 권리를 주장할 수 없습니다.
2. 당 사이트가 관계법령 및 개인정보 보호정책에 의거하여 그 책임을 지는 경우를 제외하고, 회원에게 부여된 아이디의 비밀번호 관리소홀, 부정사용 등에 의하여 발생하는 모든 결과에 대한 책임은 회원에게 있습니다.
3. 회원은 당 사이트 및 제 3자의 지식재산권을 침해해서는 안 됩니다.
4. 이용자는 당 사이트의 운영자, 직원, 기타 관계자를 사칭하는 행위를 하여서는 안 됩니다.
5. 이용자는 바이러스, 악성코드 등을 제작, 배포, 이용하여서는 아니 되고, 당 사이트의 승인 없이 광고하는 행위를 하여서는 안 됩니다.
6. 이용자는 당 사이트 및 제 3자의 명예를 훼손하거나 업무를 방해하거나, 외설적이거나, 폭력적이거나 기타 공서양속에 반하는 게시물, 쪽지, 메일 등을 게시, 전송, 배포하여서는 안 됩니다.

제 4 장 서비스의 이용

제 12 조 (서비스 이용 시간)

1. 회원의 이용신청을 승낙한 때부터 서비스를 개시합니다. 단, 일부 서비스의 경우에는 지정된 일자부터 서비스를 개시합니다.
2. 업무상 또는 기술상의 장애로 인하여 서비스를 개시하지 못하는 경우에는 사이트에 공시하거나 회원에게 이를 통지합니다.
3. 서비스의 이용은 연중무휴, 1일 24시간을 원칙으로 하며, 서비스 응대 및 처리 시간은 법정 근무일 근무시간(09:00~18:00, 법정공휴일 및 주말 제외)으로 합니다. 다만, 당 사이트의 업무상 또는 기술상의 이유로 서비스가 일시 중지될 수 있습니다. 이러한 경우 당 사이트는 사전 또는 사후에 이를 공지합니다.
4. 회원으로 가입한 이후라도 일부 서비스 이용 시 서비스 제공자의 요구에 따라 특정회원에게만 서비스를 제공할 수 있습니다.
5. 서비스를 일정범위로 분할하여 각 범위별로 이용가능 시간을 별도로 정할 수 있습니다. 이 경우 그 내용을 사전에 공개합니다.

제 13 조 (홈페이지 저작권)

1. 당 사이트가 게시한 본 홈페이지의 모든 콘텐츠에 대한 저작권은 당 사이트에 있습니다. 다만, 게시물의 원저작자가 별도로 있는 경우 그 출처를 명시하며 해당 게시물의 저작권은 원저작자에게 있습니다.
2. 회원이 직접 게시한 저작물의 저작권은 회원에게 있습니다. 다만, 회원은 당 사이트에 무료로 이용할 수 있는 권리를 허락한 것으로 봅니다.
3. 당 사이트 소유의 콘텐츠에 대하여 제3자가 허락 없이 다른 홈페이지에 사용 또는 인용

하는 것을 금지합니다.

1. 당 사이트는 기술상?운영상의 필요에 의해 제공하는 서비스의 일부 또는 전부를 변경하거나 중단할 수 있습니다. 당 사이트의 서비스를 중단하는 경우에는

30일 전에 홈페이지에 이를 공지하되, 다만 사전에 통지할 수 없는 부득이한 사정이 있는 경우는 사후에 통지를 할 수 있습니다.

2. 제1항의 경우에 당 사이트가 제공하는 서비스의 이용과 관련하여, 당 사이트는 이용자에게 발생한 어떠한 손해에 대해서도 책임을 지지 않습니다. 다만 당 사이트의 고의 또는 중대한 과실로 인하여 발생한 손해의 경우는 제외합니다.

제 5 장 계약 해지 및 이용 제한

제 15 조 (계약 해지)

1. 회원은 언제든지 마이페이지 메뉴 등을 통하여 이용계약 해지 신청을 할 수 있으며, 당 사이트는 관련법 등이 정하는 바에 따라 이를 즉시 처리하여야 합니다.

2. 회원이 계약을 해지할 경우, 관련법령 및 개인정보처리방침에 따라 당 사이트가 회원정보를 보유하는 경우를 제외하고는 해지 즉시 회원의 모든 데이터는 소멸됩니다.

3. 회원이 계약을 해지하는 경우, 회원이 작성한 게시물(공용게시판에 등록된 게시물 등)은 삭제되지 아니합니다.

제 16 조 (서비스 이용제한))

1. 당 사이트는 회원이 서비스 이용에 있어서 본 약관 및 관련 법령을 위반하거나, 다음 각 호에 해당하는 경우 서비스 이용을 제한할 수 있습니다.

- 2년 이상 서비스를 이용한 적이 없는 경우
- 기타 정상적인 서비스 운영에 방해가 될 경우

2. 상기 이용제한 규정에 따라 서비스를 이용하는 회원에게 사전 통지 후 서비스 이용을 일시정지 등 제한하거나 이용계약을 해지 할 수 있습니다. 단, 불가피한 사유로 사전 통지가 불가능한 경우에는 그러하지 아니합니다.

제 6 장 손해배상 및 기타사항

제 17 조 (손해배상)

당 사이트는 무료로 제공되는 서비스와 관련하여 회원에게 어떠한 손해가 발생하더라도 당 사이트가 고의 또는 과실로 인한 손해발생을 제외하고는 이에 대하여 책임을 부담하지 아니합니다.

제 18 조 (관할 법원)

서비스 이용으로 발생한 분쟁에 대해 소송이 제기되는 경우 민사 소송법상의 관할 법원에 제기합니다.

제 19 조 (서비스별 이용자 사전 동의 사항과 의무)) 당 사이트에 '기술정보를 제공하는 이용자는 자신의 기술정보에 대한 권리(특허권, 실용신안권, 디자인권, 상표권 등)를 법적으로 보호받기 위해서 필요한 조치를 스스로 취하여야 합니다. 당 사이트는 이용자의 권리 보장이나 취득 등을 위한 어떠한 명목의 의무나 책임도 부담하지 않고, 이를 보장하지 않으며, 이용자 개인의 행위(당 사이트의 서비스 이용 행위 일체를 포함)로 인한 어떠한 분쟁이나 어떠한 명목의 손실, 손해에 대해서도 법적 책임을 지지 아니합니다.

제 7 장 " 단체표준종합정보센터" 게시물 운영정책

제 20 조 (운영정책)

"단체표준종합정보센터" 각종 게시물에 대한 운영정책은 방송통신심의위원회의 정보통신에 관한 심의규정에 기반하며 이를 위반할 경우, "단체표준종합정보센터"

운영정책에 의해 관련 게시물은 예고 없이 삭제, 이동될 수 있으며, 게시자(회원)는 아이디 이용제한 등 "단체표준종합정보센터" 이용에 제한을 받을 수 있습니다.

제 21 조 (게시물 등록)

게시물은 실명을 통해 등록합니다. 악성 글 등을 업로드 하실 경우 게시물 삭제 또는 게시

자(회원)의 이용제한 등의 경고 조치가 가능합니다.

1. 주민등록번호 도용 타인의 개인정보를 이용한 활동이 발견될 경우, 해당 회원은 이용에 제한을 받을 수 있으며, 해당 계정은 본인인증 후에 정상적인 이용이 가능합니다. 또한 여러 개의 아이디를 생성하여 허위신고를 하거나, 타인 사칭을 통한 문제 발생 시, 그에 따른 이용 제한을 받을 수 있습니다.

제 22 조 (게시물의 저작권)

1. 게시물은 회원이 서비스를 이용하면서 게재한 글, 사진, 파일, 관련 링크 및 댓글 등을 말합니다.

2. 회원이 서비스에 등록하는 게시물로 인하여 본인 또는 타인에게 손해 및 기타 문제가 발생하는 경우, 회원은 이에 대한 책임을 질수 있으며 또한 명예훼손이나 개인정보 유출, 저작권과 같은 법률에 위배되는 게시물 및 댓글은 관련 법안에 따라 민형사상 처벌을 받을 수 있으니 이 점 유의하여 주시기 바랍니다.

제 23 조 (게시물 제한규정(삭제 및 이동))

1. 욕설/비속어 및 분란을 조장하는 게시물- 욕설 및 비속어가 담겨있거나, 연상시키는 내용- 이유 없이 회원 간의 분란을 발생시킬 여지가 있는 내용의 게시물 또는 댓글

2. 게시판 및 자료실과 관련 없는 게시물- 각 주제 분야별로 나뉘어 있는 게시판 및 자료실의 주제와 관련 없는 내용

3. 상업성 광고 및 홍보 글에 관한 게시물

4. 개인정보의 유포에 관한 게시물- 타인, 혹은 본인의 메일주소/실명/사진/전화번호/주민등록번호 등의 개인정보 또는 해당 정보가 담겨 있는 내용

5. 확인되지 않은 소문의 유포에 관한 게시물- 공개되었을 경우, 당사자의 권리침해가 우려되는 내용

6. 정치적 견해 차이 및 인종/성별/지역/종교에 대한 차별, 비하하는 게시물

- 인종/성별/지역/종교에 대한 차별적 발언 또는 비하하는 내용

- 상이한 정치적 견해를 비하하거나 폄하하는 내용

7. 타인을 사칭하거나 범죄행위에 관한 게시물

- 공인이나 특정이슈와 관련한 당사자 또는 지인, 주변인 등을 사칭하여 게시물을 작성하거나, "단체표준중합정보센터" 운영자를 사칭하여 작성된 내용

- 범죄와 관련 있거나 범죄를 유도하는 행위를 포함하는 내용

8. 저작권 위반에 관한 게시물

- 기사, 사진, 동영상, 음원, 영상물 등 저작권에 의해 보호받는 콘텐츠와 관련된 내용 뉴스의 경우, 기사 내용의 전부 혹은 일부를 게시하는 경우에는

저작권에 저촉될 수 있기 때문에 링크(URL)만을 허용합니다.

- 음원, 사진, 동영상 등 해당 자료에 대한 불법공유를 목적으로 작성한 내용 공유를 목적으로 이메일을 수집하는 행위도 동일하게 처리됩니다.

9. 악성코드/스파이웨어/협오감 조성에 관한 게시물

- 악성코드 및 스파이웨어의 유포 및 유도 관련 게시물은 사전경고 없이 제재를 받을 수 있습니다.

- 시각 및 청각적으로 타인에게 혐오감을 줄 수 있는 게시물은 사전경고 없이 제재를 받을 수 있습니다.

10. 기타 서비스 운영에 지장을 초래할 수 있는 게시물

제 24 조 (이용제한)

1. 게시물 제한규정(3조)에 해당하는 내용을 게재하는 경우

2. 다량의 게시물 등록을 목적으로 의미 없는 제목을 사용하거나, 반복되는 제목을 사용하여 게재하는 경우

3. 비정상적인 방법으로 게시물을 등록, 조회 및 추천하는 경우 등

제 25 조 (게시중단 요청 서비스)

다른 회원의 게시물로 인하여 명예훼손, 저작권 침해 등의 피해를 입었을 경우, 운영담당자를 통해 해당 게시물에 대한 게시중단을 요청하실 수 있습니다.

[부 칙]

(시행일) 이 약관은 2016년 12월 30일부터 적용되며, 종전 약관은 본 약관으로 대체되며, 개정된 약관의 적용일 이전 가입자도 개정된 약관의 적용을 받습니다.

```

    </div>
    <div class="btn_fr"><input
type="checkbox" id="ck_item1" name="ck_item1" required><label
for="ck_item1">약관의 동의</label>
    </div>
</div>
<div class="register_info">
    <input
type="text" name="address1" id="address1" placeholder="기본 주소"
"class="form-control" required /><br>
    <input
type="text" name="address2" id="address2" placeholder="상세 주소"
"class="form-control" required /><br>
    <input
type="text" name="postcode" id="postcode" style="width:160px;float:left;margin-
right:20px;" placeholder="우편번호" class="form-control">
    <button
type="button" id="post_btn" onclick="findAddr()" class="btn btn-primary">우편번
호 검색</button>
    </div>

    <div class="btn1">
    <button class="submit btn
btn-primary" type="submit">가입 완료</button>
    <button class="reset btn" type="reset">가입 취소
</button>
    </div>

    </form>
</div>
<script>
$(document).ready(function(){
$("#id").keyup(function(){
$("#idck").val("no");
if($("#this").val()!=""){
$("#msg").html("<strong>아이디 입력중입니다.</strong>");
$("#id").focus();
} else{
$("#msg").html("아직 아이디 중복 체크를 하지 않으셨습니다.");
}
});
});

function idCheck(){
if($("#id").val()==""){
```

```

alert("아이디를 입력하지 않았습니다.");
$("#id").focus();

        return false;
    }

        varparams = { id:$("#id").val() }

$.ajax({
url:"${path }/IdCheck.do",
type:"post",
dataType:"json",
data:params,
success:function(data){    //console.log(data.result);
                            varidPass = data.result;
                            if(idPass==false){

$("#idck").val("no");
$("#msg").html("<strong style='color:red'>기존에 사용되고 있는 아이디입니다. 다시 입력하시기 바랍니다.</strong>");
$("#id").focus();
} else if(idPass==true){
$("#idck").val("yes");
$("#msg").html("<strong style='color:red'>사용 가능한 아이디입니다.</strong>");
} else if(idPass==""){
$("#msg").text("<strong style='color:blue'>아이디가 확인되지 않았습니다. 다시 시도하시기 바랍니다.</strong>");
}
}
});
}

</script>
<script>
    functionjoinCheck(f){
        if(f.pw.value!=f.pw2.value){
alert("비밀번호와 비밀번호 확인이 서로 다릅니다.");
f.pw.focus();

        return false;
    }

        if(f.idck.value!="yes"){
alert("아이디 중복 체크를 하지 않으셨습니다.");
        return false;
    }
}

    functionfindAddr(){
        newdaum.Postcode({
oncomplete:function(data){
console.log(data);

        varroadAddr = data.roadAddress;
        varjibunAddr = data.jibunAddress;
document.getElementById("postcode").value = data.zonecode;
        if(roadAddr != ' '){
document.getElementById("address1").value = roadAddr;
} else if(jibunAddr != ' '){

```

```

document.getElementById("address1").value = jibunAddr;
}
document.getElementById("address2").focus();
}
}).open();
}

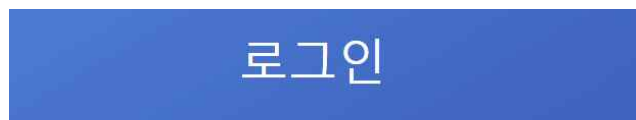
</script>

<script
src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></
script>

</div>
</section>
</div>
</div>
<footer class="ft" id="ft">
<%@ include file="../footer.jsp" %>
</footer>
</div>
</body>
</html>

```

로그인



login.jsp

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

```



```

<%@ include file="../../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
                                                                    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
    <!-- 플러그인 연결-->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet"href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet"href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet"href="${path}/css/common.css">
    <link rel="stylesheet"href="${path}/css/hd.css">
    <link rel="stylesheet"href="${path}/css/ft.css">
    <style>
        /* 본문 영역 스타일 */
        .contents{
            clear: both;
            min-height: 120vh;
            background-image: url("${path}/img/login.jpg"); /*여기 이런식으로
처리*/
            background-repeat: no-repeat;
            background-position: center -250px;
        }
        .contents::after{
            content: "";
            clear: both;
            display: block;
            width: 100%;
        }
        .page{
            clear: both;
            width: 100vw;
            height: 100vh;
            position: relative;
        }
        .page::after{
            content: "";
            display: block;
            width: 100%;
            clear: both;
        }
        .page_wrap{
            clear: both;
            width: 1200px;
            height: auto;
            margin: 0 auto;
        }
        .page_tit{

```

```

        font-size: 48px;
        text-align: center;
        padding-top: 0.7em;
        color: #fff;
        padding-bottom: 1.3em;
    }

    .breadcrumb{
        clear: both;
        width: 1200px;
        margin: 0 auto;
        text-align: right;
        color: #fff;
        padding-top: 28px;
    }

    .breadcrumb a{
        color: #fff;
    }


    .login-form{
        /*margin-top: 50px;*/
        margin: 50px auto;
        width: 350px;
        height: 400px;
        border: 0;
        border-radius: 5px;
        background-color: #eaf4fa;
        padding: 30px;
        box-shadow: 0 0 25px rgba(0, 0, 0, 0.2);
        text-align: center;
    }

    .login-form a{
        text-decoration: none;
        color: #9B9B9B;
        font-size: 18px;
        /*margin-left: 130px;*/
    }

    .tb1{
        margin-left: 22px;
    }

    .logo-wrap{
        margin: 0 auto;
        padding: 50px 10px 60px 20px;
    }

    .indata{
        font-size: 14px;
        box-sizing: border-box;
        margin-left: 15px;
        margin-bottom: 16px;
        border: none;
    }

```



```

                                required></td>
                                </tr>
                                <tr>
                                    <td>
                                        <input type="submit" value="로그인"
"class="inbtn">
                                </td>
                                </tr>
                            </tbody>
                        </table>
                    </form>
                    <a href="/member/term.jsp">회원가입 </a>
                </div>
            </div>
        </section>
    </div>
    <footer class="ft" id="ft">
<%@ include file="../../footer.jsp" %>
    </footer>
</div>
</body>
</html>

```

마이페이지

마이페이지

이름	관리자
아이디	admin
전화번호	01011112222
이메일	admin@shop.com
생일	1995-07-31
주소	서울시 구로구 01111
포인트	0

회원정보수정

메인화면

mypage.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>마이 페이지</title>
<%@ include file="../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결-->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet" href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet" href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet" href="${path}/css/common.css">
    <link rel="stylesheet" href="${path}/css/hd.css">
    <link rel="stylesheet" href="${path}/css/ft.css">
    <style>
        /* 본문 영역 스타일 */
        .contentsclear:both; min-height: 100vh; background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
    }
    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 100vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
        .page_titfont-size:48px; text-align: center; padding-top:0.7em;
color:#fff;
        padding-bottom: 1.3em; }
    .breadcrumbclear:both;
        width:1200px; margin: 0 auto; text-align: right; color:#fff;
        padding-top: 28px;}
    .breadcrumb acolor:#fff; }
    .frmclear:both; width:1200px; margin:0 auto; padding-top: 80px; }
    .tb1{
        width:600px;
        margin:0 auto;
        font-size: 20px;
```

```

        border-collapse: separate;
        border-spacing: 0 20px;
        background-color: #e1e1e1;
        border-radius: 14px;
    }

    .tb1 th{
        width:180px;
        line-height: 30px;
        padding-left: 70px;
        border-right: 2px solid #6b6b6b;
        box-sizing: border-box;
        text-align: left;
    }

    .tb1 .data{
        width:320px;
        line-height: 30px;
        padding-top:4px;
        padding-bottom:4px;
        /*border-bottom: 1px solid #333; */
        box-sizing: border-box;
        text-align: left;
        padding-left: 90px;
    }

    .tb2{
        width: 600px;
        margin: 0 auto;
        margin-top: 50px;
    }

    .inbtndisplay:block;
    border-radius:10px;
    min-width:120px;
    padding-left: 24px;
    padding-right: 24px;
    text-align: center;
    line-height: 38px;
    background-color: #333;
    color:#fff;
    font-size: 18px;
    cursor: pointer; }
    .inbtn:first-childfloat:left; }
    .inbtn:last-childfloat:right; }
    .inbtn:hover{
        background-color: #666666;
    }

</style>
<c:set var="path"value="<%=request.getContextPath() %>"/>
</head>
<body>
<div class="wrap">
    <header class="hd"id="hd">

```

```

<%@ include file="../../header.jsp" %>
</header>
<div class="contents" id="contents">
    <div class="breadcrumb">
        <p><a href="">HOME</a>> <span>마이 페이지</span></p>
    </div>
    <section class="page" id="page1">
        <div class="page_wrap">
            <h2 class="page_tit">마이 페이지</h2>
            <table class="tb1">
                <tbody>
                    <tr>
                        <th>이름</th>
                        <td class="data">${cus.name}</td>
                    </tr>
                    <tr>
                        <th>아이디</th>
                        <td class="data">${cus.id}</td>
                    </tr>
                    <tr>
                        <th>전화번호</th>
                        <td class="data">${cus.tel}</td>
                    </tr>
                    <tr>
                        <th>이메일</th>
                        <td class="data">${cus.email}</td>
                    </tr>
                    <tr>
                        <th>생일</th>
                        <td class="data">${cus.birth}</td>
                    </tr>
                    <tr>
                        <th>주소</th>
                        <td class="data">${cus.address}</td>
                    </tr>
                    <tr>
                        <th>포인트</th>
                        <td class="data">${cus.point}</td>
                    </tr>
                </tbody>
            </table>
            <table class="tb2">
                <tr>
                    <td colspan="2">
                        <a href="${path}/MypageModify.do?sid=${cus.id}" class="inbtn">회원정보수정</a>
                        <a href="/" class="inbtn">메인화면</a>
                    </td>
                </tr>
            </table>

```

```
        </div>
    </section>
</div>
<footer class="ft" id="ft">
<%@ include file="../footer.jsp" %>
</footer>
</div>
</body>
</html>
```

회원정보수정

정보수정

아이디

admin

기존 비밀번호

비밀번호 (문자, 숫자, 특수문자 포함 8~20자)

새로운 비밀번호

비밀번호 확인

새로운 비밀번호 확인

비밀번호 확인

이름

관리자

생년월일

1995-07-31

전화번호

01011112222

이메일 주소

admin@shop.com

서울시 구로구 01111

상세 주소 입력

우편번호

우편번호 검색

정보 수정

메인으로

mypageModify.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>정보수정</title>
<%@ include file="../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결-->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet" href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet" href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet" href="${path}/css/common.css">
    <link rel="stylesheet" href="${path}/css/hd.css">
    <link rel="stylesheet" href="${path}/css/ft.css">
<style>
    .contents{
        clear: both;
        min-height: 250vh;
        background-image: url("${path}/img/login.jpg");
        background-repeat: no-repeat;
        background-position: center -250px;
    }

    .contents::after{
        content: "";
        clear: both;
        display: block;
        width: 100%;
    }

    .page{
        clear: both;
        width: 100vw;
        height: 150vh;
        position: relative;
    }
}
```

```

.page::after{
    content: "";
    display: block;
    width: 100%;
    clear: both;
}

.page_wrap{
    clear: both;
    width: 1200px;
    height: auto;
    margin: 0 auto;
}

.page_tit{
    font-size: 48px;
    text-align: center;
    padding-top: 1em;
    color: #fff;
    padding-bottom: 2.4rem;
}

.breadcrumb{
    clear: both;
    width: 1200px;
    margin: 0 auto;
    text-align: right;
    color: #fff;
    padding-top: 28px;
}

.breadcrumb a{
    color: #fff;
}

d {
    box-sizing: border-box;
    font-family: 'Noto Sans KR', sans-serif;
}

.join_form{
    margin: 0 auto;
    margin-top: 50px;
    background: #eaf4fa;
    border-radius: 5px;
    width: 600px;
    height: auto;
    max-width: 100%;
    overflow: hidden;
    box-shadow: 0 0 25px rgba(0, 0, 0, 0.2);
}

.header{
    padding: 40px 40px;
    text-align: center;
}

```

```

}

.header img{
    display: inline-block;
}

.header h2{
    display: inline-block;
    margin: 0;
    font-size: 30px;
    font-weight: bolder;
    padding-bottom: 15px;
    padding-left: 5px;
}

.frm1{
    padding: 30px 40px;
}

forminf {
    padding-bottom: 10px;
}

.register_info_id{
    position: relative;
}

.register_info_id button{
    position: absolute;
    width: 90px;
    height: 40px;
    top: 0;
    bottom: 0;
    right: 5px;
    margin: 32px 0;
    border-radius: 3px;
    background: cornflowerblue;
    color: #fff;
    border: 0;
}

.register_info input{
    border: 1.5px solid #535053;
    border-radius: 3px;
    display: block;
    font-size: 11px;
    padding: 5px;
    width: 80%;
}

```

```
.submit{
  background-color: cornflowerblue;
  border: 2px solid cornflowerblue;
  width: 120px;
  border-radius: 5px;
  padding: 5px;
  color: #ffffff;
  float: left;
}
```

```
.agree_fr{
  margin: 20px auto;
  border: 1px solid #eee;
  padding: 20px;
  overflow-y: auto;
  height: 100px;
  white-space: pre-wrap;
}
```

```
.agree_fr2{
  border-radius: 5px;
  height: 100px;
  border: 1px solid #d9d6d6;
  background-color: white;
  white-space: pre-line;
  overflow-y: scroll;
  margin: 0 auto 20px;
  padding: 10px;
}
```

```
.frm1 p{
  text-align: right;
}
```

```
.frm1 input{
  display: inline-block;
  width: 100%;
  border: 1px solid #d9d6d6;
  padding: 17px 25px;
  box-sizing: border-box;
  margin: 0;
  border-radius: 5px;
}
```

```
.register_info{
  padding: 10px 0;
}
```

```

input::placeholder{
    font-size: 15px;
    color: #9d9d9d;
}

label{
    padding-bottom: 10px;
    display: block;
    font-weight: bold;
    font-size: 16px;
    color: #000;
}

label > span{
    font-weight: normal;
    color: red;
    font-size: 13px;
    padding-left: 10px;
}

.btn_fr input{
    display: inline-block;
    width: auto;
    float: left;
}

.btn_fr label{
    padding-left: 25px;
}

input.inbtn{
}

button.submit{
    display: block;
    height: 50px;
    background: cornflowerblue;
    border: 0;
    border-radius: 5px;
    width: 50%;
    color: #fff;
    font-size: 18px;
    float: left;
    margin: 0 5px;
    padding: 0;
    box-shadow: 2px 3px 7px 0px #0000005e;
}

button.reset{
    height: 50px;
    background: gray;
    border: 0;
    border-radius: 5px;
    width: 50%;
    color: #fff;
    font-size: 18px;
    padding: 0;
}

```

```

margin: 0 5px;
box-shadow: 2px 3px 7px 0px #0000005e;
}

.btn1{
color: #fff;
width: 70%;
display: flex;
margin: 20px auto;
}

</style>
<c:set var="path" value="%=request.getContextPath() %>"/>
</head>
<body>
<div class="wrap">
<header class="hd" id="hd">
<%@ include file="../header.jsp" %>
</header>
<div class="contents" id="contents">
<div class="breadcrumb">
<p><a href="">HOME</a>> <span>정보수정</span></p>
</div>
<section class="page" id="page1">
<div class="page_wrap">
<h2 class="page_tit">정보수정</h2>
<div class="join_form">
<form class="frm1" action="${path
}/MypageModifyPro.do" method="post" onsubmit="return joinCheck(this)">
<div class="register_info">
<label>아이디<!--<span>20자 이내 비밀번호를 입력하세
요</span>--></label>
<input type="text" name="id" id="id" class="indata"
placeholder="비밀번호 (문자, 숫자, 특수문자 포
함 8~20자)" value="${cus.id}" readonly>
</div>
<div class="register_info">
<label>기존 비밀번호<!--<span>20자 이내 비밀번호를 입
력하세요</span>--></label>
<input
type="password" name="pw" id="pw" class="indata"
placeholder="비밀번호 (문자, 숫자, 특수문자 포
함 8~20자)" required="">
</div>
<div class="register_info">
<label>새로운 비밀번호<!--<span>비밀번호가 일치하지
않습니다</span>--></label>
<input
type="password" name="pw2" id="pw2" class="indata" placeholder="비밀번호
확인"
required="">
</div>
<div class="register_info">

```

```

<label>새로운 비밀번호 확인<!--<span>비밀번호가 일치
하지 않습니다</span>--></label>
<input
type="password"name="pw3"id="pw3"class="indata"placeholder="비밀번호
"required="">
</div>
<div class="register_info">
<label>이름</label>
<input
type="text"name="name"id="name"class="indata"placeholder="이름
"value="{cus.name}"readonly>
</div>
<div class="register_info">
<label>생년월일</label>
<td><input
type="date"name="birth"id="birth"placeholder="생년월일
"class="form-control"value="{cus.birth}"required></td>
</div>
<div class="register_info">
<label>전화번호</label>
<input
type="tel"name="tel"id="tel"class="indata"placeholder="전화번호
"value="{cus.tel}"required="">
</div>
<div class="register_info">
<label>이메일 주소</label>
<input
type="email"name="email"id="email"class="indata"placeholder="이메일
"value="{cus.email}"required="">
</div>
<div class="register_info">
<input
type="text"name="address1"id="address1"placeholder="기본 주소
"class="form-control"value="{c:out value='{fn:split(cus.address,
"<br>")}[0]}' />" readonly required><br>
<input
type="text"name="address2"id="address2"placeholder="상세 주소
"class="form-control"value="{c:out value='{fn:split(cus.address,
"<br>")}[1]}' />" required ><br>
<input
type="text"name="postcode"id="postcode"style="width:160px;float:left;margin-
right:20px;"placeholder="우편번호"class="form-control"value="{c:out
value='{fn:split(cus.address, "<br>")}[2]}' />" readonly required>
<button
type="button"id="post_btn"onclick="findAddr()"class="btn btn-primary">우편번
호 검색</button>
</div>
<div class="btn1">
<button class="submit btn
btn-primary"type="submit">정보 수정</button>

```

<button class="reset btn" type="reset">메인으로

</button>

</div>

</form>

</div>

</div>

<script>

function findAddr(){

new daum.Postcode({

oncomplete: function(data){

console.log(data);

var roadAddr = data.roadAddress;

var jibunAddr = data.jibunAddress;

document.getElementById("postcode").value = data.zonecode;

if(roadAddr !== ''){

document.getElementById("address1").value = roadAddr;

} else if(jibunAddr !== ''){

document.getElementById("address1").value = jibunAddr;

}

document.getElementById("address2").focus();

}

}

</script>

<script

src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></

script>

</section>

</div>

<footer class="ft" id="ft">

<%@ include file="../footer.jsp" %>

</footer>

</div>

</body>

</html>

로그아웃

package shop.controller.custom;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

@WebServlet("/Logout.do")

public class LogoutCtrl extends HttpServlet {

@Override

protected

void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException{

HttpSession session = request.getSession();


```

session.invalidate();
response.sendRedirect("/pro02");
}
}

```

2. 게시판 관련 뷰

HOME > 공지사항

공지사항

| 번호 | 제목 | 작성일 | 조회수 |
|--------------|----|-----|-----|
| 작성된 글이 없습니다. | | | |

Search:

Previous
Next

글 작성

noticeList.jsp

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>공지사항</title>
<%@ include file="../head.jsp" %>
<!-- 스타일 초기화 : reset.css 또는 normalize.css -->
<link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
<!-- 플러그인 연결-->
<!-- jquery -->
<script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
<!-- datatables -->
<link
rel="stylesheet" href="https://cdn.datatables.net/1.13.6/css/jquery.dataTables
s.css">
<script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
<!-- 스타일 초기화 -->
<link rel="stylesheet" href="${path}/css/reset.css">
<!-- 웹 폰트 -->

```

```

<link rel="stylesheet"href="${path}/css/font.css">
<!-- css 모듈화 -->
<link rel="stylesheet"href="${path}/css/common.css">
<link rel="stylesheet"href="${path}/css/hd.css">
<link rel="stylesheet"href="${path}/css/ft.css">
<style>
    /* 본문 영역 스타일 */
    .contents{
        clear: both;
        min-height: 100vh;
        background-image: url("${path}/img/login.jpg");
        background-repeat: no-repeat;
        background-position: center -250px;
    }

    .contents::after{
        content: "";
        clear: both;
        display: block;
        width: 100%;
    }

    .page{
        clear: both;
        width: 100vw;
        height: 100vh;
        position: relative;
    }

    .page::after{
        content: "";
        display: block;
        width: 100%;
        clear: both;
    }

    .page_wrap{
        clear: both;
        width: 1200px;
        height: auto;
        margin: 0 auto;
    }

    .page_tit{
        font-size: 48px;
        text-align: center;
        /*padding-top: 0.7em;*/
        color: #fff;
        padding-bottom: 1.3em;
    }

    .breadcrumb{
        clear: both;
        width: 1200px;
        margin: 0 auto;
        text-align: right;
    }

```

```

        color: #fff;
        padding-top: 28px;
        padding-bottom: 28px;
    }

    .breadcrumb a{
        color: #fff;
    }

    /* 테이블 스타일 */
    .tb1{
        width: 80%;
        margin: 50px auto;
        font-size: 20px;
        border-collapse: collapse;
    }

    .tb1 th{
        background-color: #527AF2;
        color: #fff;
        padding: 16px;
        border: 1px solid #527AF2;
    }

    .tb1 td{
        padding: 12px 16px;
        border: 1px solid #ddd;
        text-align: center;
        line-height: 24px;
    }

    .tb1 th:first-child{
        width: 40px;
    }

    .tb1 tbody{
    }

    .tb1 .item1{
        width: 8%;
    }

    .tb1 .item2{
        width: 50%;
        text-align: left;
    }

    .tb1 .item3{
        width: 12%;
    }

    /* 기타 버튼 스타일 */
    .inbtn{
        display: block;
        border-radius: 10px;
        min-width: 60px;
        padding-left: 24px;
        padding-right: 24px;
        text-align: center;
        line-height: 38px;
    }

```


공지사항 작성

글 작성

내용

글쓰기

취소

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
  <c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>게시글 작성</title>
<%@ include file="/head.jsp" %>
  <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
  <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
  <!-- 플러그인 연결-->
  <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
```

```

<!-- 스타일 초기화 -->
<!-- 스타일 초기화 -->
<link rel="stylesheet"href="${path}/css/reset.css">
<!-- 웹 폰트 -->
<link rel="stylesheet"href="${path}/css/font.css">
<!-- css 모듈화 -->
<link rel="stylesheet"href="${path}/css/common.css">
<link rel="stylesheet"href="${path}/css/hd.css">
<link rel="stylesheet"href="${path}/css/ft.css">
<style>
    /* 본문 영역 스타일 */
        .contentsclear:both; min-height: 100vh; background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
}

    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 100vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
    .page_titfont-size:48px; text-align: center; color:#fff;
padding-bottom: 1.3em; }
    .breadcrumbclear:both;
width:1200px; margin: 0 auto; text-align: right; color:#fff;
padding-top: 28px; padding-bottom: 28px; }
    .breadcrumbb acolor:#fff; }
    .tb1width:600px; margin:0 auto; font-size: 24px;}
        .tb1 thwidth: 200px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tb1 tdwidth: 400px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tb1 td:last-childline-height: 48px; padding-top:24px;
padding-bottom:24px; }

    .inbtndisplay:block;
border-radius:10px;
min-width:120px;
padding-left: 24px;
padding-right: 24px;
text-align: center;
line-height: 38px;
background-color: #333;
color:#fff;
font-size: 18px;
cursor: pointer; }
    .inbtn:first-childfloat:left; }

```

[illegible]


```

name="content" id="content" cols="50" rows="15" placeholder="내용"
"required"></textarea></td>
</tr>
<tr>
<td colspan="2">
<input type="submit" value="글쓰기"
"class="inbtn">
<input type="reset" value="취소"
"class="inbtn" onclick="window.history.back();">
</td>
</tr>
</tbody>
</table>
</form>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="/footer.jsp" %>
</footer>
</div>
</body>
</html>

```

글 내용 수정

글 작성 수정

내용 수정

글쓰기
취소

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="%=request.getContextPath() %=" />
<!DOCTYPE html>

```

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<%@ include file="/head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
                                                                    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결-->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- 스타일 초기화 -->
    <!-- 스타일 초기화 -->
    <link rel="stylesheet" href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet" href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet" href="${path}/css/common.css">
    <link rel="stylesheet" href="${path}/css/hd.css">
    <link rel="stylesheet" href="${path}/css/ft.css">
    <style>
        /* 본문 영역 스타일 */
        .contentsclear:both; min-height: 100vh; background-image:
url("../img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
    }
    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 100vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
    .page_titfont-size:48px; text-align: center; color:#fff;
        padding-bottom: 1.3em; }
    .breadcrumbclear:both;
        width:1200px; margin: 0 auto; text-align: right; color:#fff;
        padding-top: 28px; padding-bottom: 28px; }
    .breadcrumbb acolor:#fff; }
    .tblwidth:600px; margin:0 auto; font-size: 24px;}
        .tbl thwidth: 200px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tbl tdwidth: 400px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tbl td:last-childline-height: 48px; padding-top:24px;
padding-bottom:24px; }

    .inbtndisplay:block;

```



```

<form
action="${path}/UpdateNoticePro.do" id="login_frm" class="frm" method="post">
    <table class="tb1">
        <tbody>
            <tr>
                <td colspan="2">
                    <input
type="text" name="title" id="title" class="indata" placeholder="제목"
"value="${noti.title}" autofocus required>
                    <input
type="hidden" name="no" id="no" value="${noti.no}" readonly>
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <td colspan="2"><textarea
name="content" id="content" cols="50" rows="15" placeholder="내용"
"required">${noti.content}</textarea></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <input type="submit" value="글쓰기"
"class="inbtn">
                            <input type="reset" value="취소"
"class="inbtn" onclick="window.history.back();">
                        </td>
                    </tr>
                </tbody>
            </table>
        </form>
    </div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="/footer.jsp" %>
</footer>
</div>
</body>
</html>

```

게시글

글 작성 수정

수정 삭제

관리자

23-09-04

조회수 2

내용 수정

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.List" %>
<%@ page import="java.text.*" %>
<%@ page import="java.util.Date" %>
<%@ page import="shop.dto.Notice" %>
<%@ page import="shop.model.DBConnect" %>
<%@ page import="shop.model.MariaDBCon" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>자유 게시판 상세</title>
<%@ include file="../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결 -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- datatables -->
    <link
rel="stylesheet" href="https://cdn.datatables.net/1.13.6/css/jquery.dataTable
s.css">
```

```

<script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
<!-- 스타일 초기화 -->
<link rel="stylesheet"href="${path}/css/reset.css">
<!-- 웹 폰트 -->
<link rel="stylesheet"href="${path}/css/font.css">
<!-- css 모듈화 -->
<link rel="stylesheet"href="${path}/css/common.css">
<link rel="stylesheet"href="${path}/css/hd.css">
<link rel="stylesheet"href="${path}/css/ft.css">
<style>
    /* 본문 영역 스타일 */
        .contentsclear:both;    min-height:    180vh;    background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
}

    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 100vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
    .page_titfont-size:48px; text-align: center; color:#fff;
        padding-bottom: 1.3em; }
    .breadcrumbclear:both;
        width:1200px; margin: 0 auto; text-align: right; color:#fff;
        padding-top: 28px; padding-bottom: 28px; }
    .breadcrumbb acolor:#fff; }
    .tb1{
        width: 1200px;
        font-size: 17px;
        margin-bottom: 50px;
        border-collapse: collapse; /* 테이블 간격 없애기 */
}

    .tb1 thead th{
        line-height: 32px;
        padding: 12px 15px; /
        /*border: 1px solid #f5f5f5; !*/
        box-sizing: border-box;
        background-color: #eeeeee; /* 배경색 조정 */
        font-size: 22px;
        font-weight: 600;
        border-radius: 10px;
}

    .tb1 thead td{
        font-size: 16px;
        text-align: right;
        line-height: 40px;
        border-bottom: #7e7e7e 1px solid;
}

    .tb1 thead td:nth-child(1) {
        width: 6%;

```



```

.tb2 thead tr th{
    text-align: center;
}

.tb2 tbody tr td:first-child{
    text-align: center;
    font-weight: bold;
}

.tb2 tbody tr td:nth-child(2){
    word-break: break-word;
}

.tb2 tbody tr td:nth-child(3){
    text-align: center;
}

.inbtn, .delete_btn{
    display: inline-block;
    border-radius: 5px;
    width: 50px;
    line-height: 28px;
    text-align: center;
    font-size: 14px;
    cursor: pointer;
    transition: background-color 0.3s; /* 마우스 오버 효과 */
}

.inbtn{
    background-color: #333;
    color: #fff;
}

.delete_btn{
    background-color: red;
    color: #fff;
}

.inbtn:hover, .delete_btn:hover{
    background-color: #555;
}

.frm{
    margin-top: 50px;
}

/*.frm tr > * {*/
/*    margin-right: 10px;*/
/*}*/

.btn_group{
    margin-top: 50px;
}

#ans_btn{
    border-radius: 5px;
    width: 80px; /* 버튼 크기 */
    padding: 7px 15px;
    text-align: center;
}

```



```

</a></p>
</div>
<section class="page" id="page1">
  <div class="page_wrap">
    <h2 class="page_tit">게시글</h2>
    <table class="tbl">
      <thead>
        <tr class="title">
          <th colspan="5">${noti.title}</th>
        </tr>
        <tr>
          <td>
            <c:if test="${sid eq 'admin' }">
              <a
href="${path}/UpdateNotice.do?no=${noti.no}" class="inbtn">수정</a>
            </c:if>
          </td>
          <td>
            <c:if test="${sid eq 'admin' }">
              <a
href="${path}/DeleteNotice.do?no=${noti.no}" class="inbtn delete_btn">삭제</a>
            </c:if>
          </td>
          <td>관리자</td>
          <td>${noti.resdate}</td>
          <td>조회수 ${noti.visited}</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td colspan="5" class="content">
            <div>${noti.content}</div>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="../footer.jsp" %>
</footer>
</div>
</body>
</html>

```

게시글 삭제

```

package shop.controller.notice;
import shop.model.NoticeDAO;
import javax.servlet.ServletException;

```

```

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/DeleteNotice.do")
public class DeleteNoticeCtrl extends HttpServlet {
    @Override
        protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException{
            int no = Integer.parseInt(request.getParameter("no"));
            NoticeDAO dao = new NoticeDAO();
            int a = dao.deleteNotice(no);
            PrintWriter out = response.getWriter();
            if(a>0){
response.sendRedirect("/pro02/NoticeList.do");
            } else{
out.println("<script>history.go(-1);</script>");
            }
        }
    }
}

```

상품 관련 뷰
상품 목록

정보	가격	출간일	비고
 <div>초등수학1 초등수학1 목차</div>	12000	23-09-01	상세보기 장바구니 구매하기
 <div>중등국어1 중등국어1 목차</div>	20000	23-09-01	상세보기 장바구니 구매하기

Search: Previous Next [추가 입고](#) [상품 입고](#) [상품 등록](#)

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>

```

```

<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>도서목록</title>
<%@ include file="../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
    <!-- 플러그인 연결-->
    <!-- jquery -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- datatables -->
    <link
rel="stylesheet"href="https://cdn.datatables.net/1.13.6/css/jquery.dataTable
s.css">
    <script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet"href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet"href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet"href="${path}/css/common.css">
    <link rel="stylesheet"href="${path}/css/hd.css">
    <link rel="stylesheet"href="${path}/css/ft.css">
    <style>
        /* 본문 영역 스타일 */
        .contents{
            clear: both;
            min-height: 200vh;
            background-image: url("${path}/img/login.jpg");
            background-repeat: no-repeat;
            background-position: center -250px;
        }
        .contents::after{
            content: "";
            clear: both;
            display: block;
            width: 100%;
        }
        .page{
            clear: both;
            width: 100vw;
            height: 100vh;
            position: relative;
        }
    </style>

```

```

.page::after{
    content: "";
    display: block;
    width: 100%;
    clear: both;
}

.page_wrap{
    clear: both;
    width: 1200px;
    height: auto;
    margin: 0 auto;
}

.page_tit{
    font-size: 48px;
    text-align: center;
    /*padding-top: 0.7em;*/
    color: #fff;
    padding-bottom: 1.3em;
}

.breadcrumb{
    clear: both;
    width: 1200px;
    margin: 0 auto;
    text-align: right;
    color: #fff;
    padding-top: 28px;
    padding-bottom: 28px;
}

.breadcrumb a{
    color: #fff;
}

/* 테이블 스타일 */
.tb1{
    width: 80%;
    margin: 50px auto;
    font-size: 20px;
    border-collapse: collapse;
}

.tb1 th{
    background-color: #527AF2;
    color: #fff;
    padding: 16px;
    border: 1px solid #527AF2;
}

.tb1 th:first-child{
    width: 40px;
}

#myTable td{
    max-height: 200px;
    overflow: hidden; /* Overflow 처리 (필요한 경우) */
}

```



```

        </td>
        <td>
            <div>
                <div>
                    <div>
                        <div>
                            <div>
                                <div>
                                    </div>
                                </td>
                                <td>
                                    <div>${pro.price}</div>
                                </td>
                                <td>
                                    <div>${pro.resdate}</div>
                                </td>
                                <td>
                                    <div>
                                        <a href="${path}/Product.do?no=${pro.no}" class="btn1">상세 보기</a>
                                        </div>
                                        <c:if test="${not empty sid}">
                                            <div>
                                                <a href="${path}/AddCart.do?pno=${pro.no }" class="btn1">장바구니</a>
                                                </div>
                                                <div>
                                                    <a href="${path}/AddPayment.do?pno=${pro.no }" class="btn1">구매하기</a>
                                                </div>
                                            </c:if>
                                        </td>
                                    </tr>
                                </c:forEach>
                            </tbody>
                        </table>
                        <script>
$(document).ready( function() {
$('#myTable').DataTable({
    pageLength : 5,
    order: [[3, 'desc']], // 0번째 컬럼을 기준으로 내림차순 정렬
    info: false,
    dom: 't<f>p',
    language: {
        emptyTable: '작성된 글이 없습니다.'
    }
});
});
    </script>

```



```

$(document).ready(function() {
$($('.dataTables_paginate')).css({
    'textAlign': 'left',
    'float': 'none',
    'margin-top': '10px',
});
$($('.dataTables_filter')).css({
    'float': 'left',
    'margin-top': '14px',
    'margin-right': '280px'
});
$($('#myTable_paginate')).css({
    'margin-right': '100px'
});

});

</script>
<div class="btn_group">
    <c:if test="${sid eq 'admin' }">
        <a href="${path}/AddProduct.do" class="inbtn">상품 등록
    </a>
        <a href="${path}/AddReceive.do" class="inbtn">상품 입고
    </a>
        <a href="${path}/UpdateReceive.do" class="inbtn">추가
    입고</a>
    </c:if>
</div>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="../footer.jsp" %>
</footer>
</div>
</body>
</html>

```

상품 추가

카테고리

고등

상품 이름

고등국어1

상품 설명

고등국어1

상품 목차

고등국어1

상품 가격

24000

이미지

파일 선택 고등국어1.jpg

샘플 영상

파일 선택 video2.mp4

상품등록

취소

상품목록

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
  <c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>상품 정보 작성</title>
<%@ include file="/head.jsp" %>
<!-- 스타일 초기화 : reset.css 또는 normalize.css -->
<link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
<!-- 플러그인 연결-->
<script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
<!-- 스타일 초기화 -->
<!-- 스타일 초기화 -->
<link rel="stylesheet"href="${path}/css/reset.css">
<!-- 웹 폰트 -->
<link rel="stylesheet"href="${path}/css/font.css">
<!-- css 모듈화 -->
<link rel="stylesheet"href="${path}/css/common.css">
<link rel="stylesheet"href="${path}/css/hd.css">
<link rel="stylesheet"href="${path}/css/ft.css">
<style>
/* 본문 영역 스타일 */
        .contentsclear:both; min-height: 100vh; background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
}

        .contents::aftercontent:""; clear:both; display:block; width:100%; }
        .pageclear:both; width: 100vw; height: 200vh; position:relative; }
        .page::aftercontent:""; display:block; width: 100%; clear:both; }
        .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
        .page_titfont-size:48px; text-align: center; color:#fff;
padding-bottom: 1.3em; }
        .breadcrumbclear:both;
width:1200px; margin: 0 auto; text-align: right; color:#fff;
padding-top: 28px; padding-bottom: 28px; }
        .breadcrumbb acolor:#fff; }
        .tb1width:600px; margin:0 auto; font-size: 24px;}
        .tb1 thwidth: 200px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tb1 tdwidth: 400px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tb1 td:last-childline-height: 48px; padding-top:24px;
padding-bottom:24px; }

        .inbtndisplay:block;
border-radius:10px;
min-width:120px;
padding-left: 24px;

```



```

        <tbody>
        <tr>
            <th>카테고리</th>
            <td colspan="2">
                <select
name="cate" id="cate" class="indata" autofocus required>
                    <option value="초등" selected>초등</option>
                    <option value="중등">중등</option>
                    <option value="고등">고등</option>
                </select>
            </td>
        </tr>
        <tr>
            <th>상품 이름</th>
            <td colspan="2"><input
type="text" name="pname" id="pname" class="indata" placeholder="상품 이름 입력"
required></td>
        </tr>
        <tr>
            <th>상품 설명</th>
            <td colspan="2"><textarea
name="pcomment" id="pcomment" cols="50" rows="15" placeholder="상품 설명 입력"
required></td>
        </tr>
        <tr>
            <th>상품 목차</th>
            <td colspan="2"><textarea
name="plist" id="plistt" cols="50" rows="8" placeholder="상품 목차 입력"
required></td>
        </tr>
        <tr>
            <th>상품 가격</th>
            <td colspan="2"><input
type="text" name="price" id="price" class="indata" placeholder="상품 가격 입력"
required></td>
        </tr>
        <tr>
            <th>이미지</th>
            <td colspan="2"><input
type="file" name="imgsrc1" id="imgsrc1" class="indata" placeholder="상품 이미지"
"accept=".jpg, .png" required></td>
        </tr>
        <tr>
            <th>샘플 영상</th>
            <td colspan="2"><input
type="file" name="imgsrc2" id="imgsrc2" class="indata" placeholder="상품 소개 영상"
"accept=".mp4" required></td>
        </tr>
        <td colspan="2">

```

```

<input type="submit" value="상품등록
" class="inbtn">
<input type="reset" value="취소
" class="inbtn" onclick="window.history.back();">
<a href="${path}/ProList.do" class="inbtn">상품
목록</a>
</td>
</tr>
</tbody>
</table>
</form>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="/footer.jsp" %>
</footer>
</div>
</body>
</html>

```

상품입고

상품 입고

상품명

고등국어1 ▼

수량

300 ▼

가격

24000

상품입고

취소

상품목록

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>상품 입고</title>
    <c:set var="path" value="<%=request.getContextPath() %>" />
<%@ include file="../../head.jsp"%>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결-->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- 스타일 초기화 -->
    <!-- 스타일 초기화 -->
    <link rel="stylesheet" href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet" href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet" href="${path}/css/common.css">
    <link rel="stylesheet" href="${path}/css/hd.css">
    <link rel="stylesheet" href="${path}/css/ft.css">
    <link rel="stylesheet" href="${path}/css/asidebar.css">
    <style>
        /* 본문 영역 스타일 */
        .contentsclear:both; min-height: 100vh; background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
    }

    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 200vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
    .page_titfont-size:48px; text-align: center; color:#fff;
        padding-bottom: 1.3em; }
    .breadcrumbclear:both;
        width:1200px; margin: 0 auto; text-align: right; color:#fff;
        padding-top: 28px; padding-bottom: 28px; }
    .breadcrumbb acolor:#fff; }
    .tb1width:600px; margin:0 auto; font-size: 24px;}
        .tb1 thwidth: 200px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tb1 tdwidth: 400px; line-height: 32px; padding-top:16px;

```

```
padding-bottom:16px;
        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tbl td:last-childline-height: 48px; padding-top:24px;
padding-bottom:24px; }
```

```
.inbtndisplay:block;
border-radius:10px;
min-width:120px;
padding-left: 24px;
padding-right: 24px;
text-align: center;
line-height: 38px;
background-color: #333;
color:#fff;
font-size: 18px;
cursor: pointer; }
.inbtn:first-childfloat:left; }
.inbtn:last-childfloat:right; }
.inbtn:hover{
background-color: #666666;
}
```

```
.indata{
display:inline-block;
width: 590px;
height: 32px;
line-height: 32px;
text-indent:10px;
font-size:16px;
border: 1px solid #ccc;
border-radius: 5px;
padding: 5px 15px;
}
```

```
textarea{
resize: none;
width: 600px;
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px;
font-size: 16px;
}
```

```
</style>
</head>
<body>
<div class="wrap">
    <header class="hd" id="hd">
<%@ include file="../../../header.jsp" %>
```



```

    </header>
<%--
<%@ include file="./adminAside.jsp"%>
--%>
    <div class="contents" id="contents">
        <div class="breadcrumb">
            <p><a href="/">HOME</a>> <a href="/">상품 입고</a>
&gt; <a href="/">상품 입고</a></p>
        </div>
        <section class="page" id="page1">
            <div class="page_wrap">
                <h2 class="page_tit">상품 입고</h2>
                <form name="frm1" id="frm1" action="${path
}/AddReceivePro.do" method="post">
                    <table class="tb1">
                        <tbody>
                            <tr>
                                <th><label for="pno">상품명</label></th>
                                <td>
                                    <c:set var="cate"/>
                                    <select name="no" id="pno">
                                        <option
value="1" cate="0" price="" selected>선택안함</option>
                                        <c:forEach var="pro" items="${proList
}" varStatus="status">
                                            <option value="${pro.no
}" cate="${pro.cate }" price="${pro.price }">${pro.pname }</option>
                                        </c:forEach>
                                    </select>
                                </td>
                            </tr>
                            <tr>
                                <th><label for="amount">수량</label></th>
                                <td><input
type="number" name="amount" id="amount" placeholder="입고 수량 입력
"class="form-control" value="1" min="1" max="1000" required></td>
                            </tr>
                            <tr>
                                <th><label for="price">가격</label></th>
                                <td><input
type="number" name="price" id="price" value="" readonly required></td>
                            </tr>
                            <tr>
                                <td colspan="2">
                                    <input
type="submit" name="submit-btn" class="inbtn" value="상품입고">
                                    <input
type="reset" name="reset-btn" class="inbtn" value="취소">
                                    <a href="${path
}/ProList.do" class="inbtn">상품목록</a>

```

```

        </td>
    </tr>
</tbody>
</table>
</form>
<script>
$(document).ready(function(){
$("#pno").change(function(){
    varv = $(this).val();

    varcate =
    $(this).find("option:selected").attr("cate");
    varprice =
    parseInt($(this).find("option:selected").attr("price"));
    $("#cate").text(cate);
    $("#rprice").val(price);
});
});

</script>
<!-- 상품을 선택하면 해당 상품의 가격을 이동 -->
<script>
    constselectElement = document.getElementById("pno");
    constpriceInput = document.getElementById("price");
selectElement.addEventListener("change", function() {
    constselectedOption =
selectElement.options[selectElement.selectedIndex];
    constprice = selectedOption.getAttribute("price");
priceInput.value = price;
});
</script>
</div>
</section>
</div>
<%@ include file="../../footer.jsp" %>
</div>
</body>
</html>

```

상품상세보기

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.text.*" %>
<%@ page import="java.util.Date" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>상품 상세</title>
<%@ include file="../../head.jsp" %>
<!-- 스타일 초기화 : reset.css 또는 normalize.css -->
<link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
<!-- 플러그인 연결-->
<script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
<!-- datatables -->
<link
rel="stylesheet"href="https://cdn.datatables.net/1.13.6/css/jquery.dataTables
s.css">
<script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
<!-- 스타일 초기화 -->
<link rel="stylesheet"href="${path}/css/reset.css">
<!-- 웹 폰트 -->
<link rel="stylesheet"href="${path}/css/font.css">
<!-- css 모듈화 -->
<link rel="stylesheet"href="${path}/css/common.css">
<link rel="stylesheet"href="${path}/css/hd.css">
<link rel="stylesheet"href="${path}/css/ft.css">
<link rel="stylesheet"href="${path}/css/video.css">
<link rel="stylesheet"href="${path}/css/productpage.css">
<style>
        .contentsclear:both; min-height: 150vh; background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
    }
    .contents::aftercontent:""; clear:both; display:block; width:100%; }
</style>
</head>
<body>
<div class="wrap">
    <header class="hd"id="hd">
<%@ include file="../../header.jsp" %>
    </header>
    <div class="contents"id="contents">
        <div class="breadcrumb">
            <p><a href="/">HOME</a>&gt; <a href="">도서 목록</a>&gt; <a
href="">상품 상세</a></p>
        </div>
        <section class="page"id="page1">
            <div class="page_wrap">
                <h2 class="page_tit">상품 상세</h2>
                <div class="product">
                    <div class="productimg">
                        

```

```

    </div>
    <div class="productdesc">
        <table class="productdesctable">
            <tr class="pname">
                <th>${pro.pname}</th>
            </tr>
            <tr class="pprice">
                <td>
                    ${pro.price} 원
                </td>
            </tr>
            <tr class="pcate">
                <td>
                    카테고리 : ${pro.cate}
                </td>
            </tr>
            <tr class="pdate">
                <td>
                </td>
            </tr>
            <tr class="pbtn">
                <c:if test="${not empty sid}">
                    <c:set
var="isLiked"value="${likedProductIds.contains(pro.no)}"/>
                    <td><a href="${path
}/AddPayment.do?pno=${pro.no }"class="inbtn">구매하기</a>
                    <a href="${path }/AddCart.do?pno=${pro.no
}&cate=${pro.cate}"class="inbtn">장바구니</a>
                    </td>
                </c:if>
            </tr>
            <tr>
                <td class="adminbtn">
                    <c:if test="${sid eq 'admin'}">
                        <a
href="${path}/UpdateProduct.do?no=${pro.no}"class="inbtn">수정</a>
                        <a
href="${path}/DeleteProduct.do?no=${pro.no}"class="inbtn delete_btn">삭제</a>
                    </c:if>
                </td>
            </tr>
        </table>
    </div>
</div>
<script>
    functiontoggleLike(productNo, ${sid }) {
$.ajax({
url: "ProductLike.do",
method: "POST",
data: {

```

```

pno: productNo,
sid: ${sid }
},
success: function(response) {
    var likeButton = $("[data-product-id='"+
productNo + "'"]);
    if(response.trim() === "liked") {
alert("상품을 좋아요 했습니다!");
likeButton.css("color", "#ff5050");
} else if(response.trim() === "unliked") {
alert("상품의 좋아요를 취소했습니다.");
likeButton.css("color", "#b4b4b4");
} else{
alert("오류가 발생했습니다. 다시 시도해주세요.");
}
}
});
$(document).ready(function() {
    // 좋아요 상태를 기반으로 버튼 색 변경
$(".inbtn").each(function() {
    var isLiked = $(this).hasClass("liked");
    if(isLiked) {
$(this).addClass("liked");
}
});
});
</script>

```

```

<table class="tb2">
    <thead>
        <tr class="title">
            <th colspan="5">상세 설명</th>
        </tr>
    </thead>
</table>
<table class="tb4">
    <tr>
        <td colspan="5" class="desc-area">

src="${path}/storage/${pro.imgSrc1}" width="320"/>
        <br><br><br>
        ${pro.plist}
        <br><br><br>
        ${pro.pcomment}
        <br><br><br>
        <br><br><br>
    </td>

```

```
</tr>
</table>
```

```
<c:if test="${not empty pro.imgSrc2}">
<table class="tb1">
  <thead>
    <tr class="title">
      <th colspan="5">시범 강의</th>
    </tr>
  </thead>
</table>
<div class="player">
  <div class="vdo_fr">
    <video id="video">
<source
src="${path}/storage/${pro.imgSrc2}" type="video/mp4"/>
    </video>
  </div>
  <div id="timebar">
    <span id="currentTime"></span>
  </div>
  <div id="buttonbar">
    <button
id="restart" class="player_btn"></button>
    <button
id="rew" class="player_btn"></button>
    <button
id="play" class="player_btn"></button>
    <button
id="pause" class="player_btn"></button>
    <button
id="stop" class="player_btn"></button>
    <button
id="fastFwd" class="player_btn"></button>
    <button
id="mute" class="player_btn"></button>
    <button
id="unmute" class="player_btn"></button>
    <input
id="vol" type="range" value="500" min="0" max="1000">
    <button
id="vmup" class="player_btn"></button>
    <button
id="vmdown" class="player_btn"></button>
    <button id="volTxt">100%</button>
    <button
id="sizeup" class="player_btn"></button>
    <button
id="sizedown" class="player_btn"></button>
```

```

id="full" class="player_btn"></button>
id="original" class="player_btn"></button>
    </div>
    <div id="progress">
        <div id="progressBar"></div>
    </div>
</div>
</c:if>
<script src="./js/vdo.js"></script>

<table class="tb2" id="myTable">
    <thead>
    <tr class="title">
        <th colspan="5">후기</th>
    </tr>
    <tr>
        <th class="item1">작성자</th>
        <th class="item2">댓글</th>
        <th class="item3">작성일</th>
        <th class="item4"></th>
    </tr>
    </thead>
    <tbody>
    <c:forEach var="lev" items="${revList }">
    <tr>
        <td class="item1">${lev.cid}</td>
        <td class="item2">${lev.content}</td>
        <td class="item3">${lev.resdate}</td>
        <td class="item4">
            <c:if test="${sid eq lev.cid || sid eq 'admin'}">
                <a
href="${path}/UpdateReview.do?cid=${lev.cid}&par=${pro.no}" class="inbtn">수정
</a>
                <a
href="${path}/DeleteReview.do?cid=${lev.cid}&par=${pro.no}" class="inbtn
delete_btn">삭제 </a>
            </c:if>
        </td>
    </tr>
    </c:forEach>
    </tbody>
</table>
<script>
$(document).ready( function() {
$('#myTable').DataTable({
    // sorting 화살표 제거
    "targets": 'no-sort',

```

```

        "bSort": false,
        // 3번째 컬럼을 기준으로 내림차순 정렬
order: [[3, 'asc']],
pageLength : 5,
searching: false, //검색 제거
lengthChange: false, // show entries 제거
info: false,
language: {
emptyTable: '작성된 후기가 없습니다.'
}
});
$('#myTable').css({
        'border': 'none',
});
} );

</script>
<form action="${path}/AddReview.do" id="login_frm" class="frm">
    <table class="tb3">
        <tbody>
            <tr>
                <c:choose>
                    <c:when test="${check == '2'}">
                        <th>${sid}</th>
                        <th><textarea
name="content" id="content" cols="100" rows="5" placeholder="리뷰 작성" required
></textarea></th>
                        <th><input type="submit" value="글쓰기"
" class="inbtn" id="ans_btn"></th>
                        <input
type="hidden" name="pno" value="${pro.no}" readonly>
                    </c:when>
                    <c:when test="${check == '1'}">
                        <p id="nlogin_comment">구매 확정을 해야 후
기 작성이 가능합니다.</p>
                    </c:when>
                    <c:otherwise>
                        <p id="nlogin_comment">도서를 구입한 고객님의
후기 작성이 가능합니다.</p>
                    </c:otherwise>
                </c:choose>
            </tr>
        </tbody>
    </table>
</form>

</div>
<footer class="ft" id="ft">
<%@ include file="../../../footer.jsp" %>
</footer>

```


</section>
</div>

</div>
</body>
</html>

4. 결제, 배송 뷰
구매



도서 번호	고등국어1
도서명	고등국어1
도서 설명	고등국어1
도서 목차	고등국어1
가격 (부가세 10% 별도)	26400
구매 수량 (남은 재고 : 300)	<input type="text" value="1"/>

배송 정보

배송지 주소

기본 주소 입력	우편번호 검색
상세 주소 입력	
우편번호	

받는 사람 연락처

결제 정보

결제 수단

결제사

결제 번호

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.text.*" %>
<%@ page import="java.util.Date" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
```

```

<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>상품 상세</title>
<%@ include file="../head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
    <!-- 플러그인 연결-->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- datatables -->
    <link
rel="stylesheet"href="https://cdn.datatables.net/1.13.6/css/jquery.dataTables
s.css">
    <script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet"href="${path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet"href="${path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet"href="${path}/css/common.css">
    <link rel="stylesheet"href="${path}/css/hd.css">
    <link rel="stylesheet"href="${path}/css/ft.css">
    <!-- 결제 api -->
    <script
type="text/javascript"src="https://cdn.iamport.kr/js/iamport.payment-1.1.5.j
s"></script>
    <style>
        /* 본문 영역 스타일 */
        /* background-image: url("../img/login.jpg"); */
        .contentsclear:both; min-height: 250vh; background-image:
url("${path}/img/login.jpg");
        background-repeat: no-repeat; background-position: center -250px;
    }
    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 100vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
    .page_titfont-size:48px; text-align: center; color:#fff;
        padding-bottom: 1.3em; }
    .breadcrumbclear:both;
        width:1200px; margin: 0 auto; text-align: right; color:#fff;
        padding-top: 28px; padding-bottom: 28px; }
    .breadcrumb acolor:#fff; }
    .tb1{

```

```

        width: 1200px;
        font-size: 17px;
        margin-bottom: 50px;
        border-collapse: collapse; /* 테이블 간격 없애기 */
    }

    .tb1 thead th{
        line-height: 32px;
        padding: 12px 15px; /
        /*border: 1px solid #f5f5f5; !*/
        box-sizing: border-box;
        background-color: #eeeeee; /* 배경색 조정 */
        font-size: 22px;
        font-weight: 600;
        border-radius: 10px;
    }

    .tb1 thead td{
        font-size: 16px;
        text-align: right;
        line-height: 40px;
        border-bottom: #7e7e7e 1px solid;
    }

    .tb1 thead td:nth-child(1) {
        width: 6%;
        text-align: center;
    }.tb1 thead td:nth-child(2) {
        text-align: left;
        width: 14%;
    }

    .tb1 thead td:nth-child(3) {
        font-weight: 600;
        width: 62%;
    }

    .tb1 thead td:nth-child(4) {
        width: 10%;
    }

    .tb1 thead td:nth-child(5) {
        width: 8%;
    }

    /*글 내용*/
    .tb1 tbody tr td{
        padding-left: 10px;
        padding-top: 40px;
        font-size: 18px;
        text-align: left;
    }

    .content{
        min-height: 400px;
        border-bottom: 1px solid #7e7e7e;
    }

```

```
}
```

```
.title{  
    text-align: left;  
    font-weight: bold;  
}
```

```
}
```

```
.tb2{  
    width: 1200px;  
}
```

```
}
```

```
.tb2 .item1width: 10%; }  
.tb2 .item2width: 68%;}  
.tb2 .item3width: 10%; }  
.tb2 .item4width: 12%;
```

```
}
```

```
.tb2 td{  
    border-bottom: 1px solid #7e7e7e;  
    vertical-align: middle;  
    height: 60px;  
}
```

```
}
```

```
.tb2 thead tr th{  
    text-align: center;  
}
```

```
}
```

```
.tb2 tbody tr td:first-child{  
    text-align: center;  
    font-weight: bold;  
}
```

```
}
```

```
.tb2 tbody tr td:nth-child(2){  
    word-break: break-word;  
}
```

```
}
```

```
.tb2 tbody tr td:nth-child(3){  
    text-align: center;  
}
```

```
}
```

```
.btn1{  
    display: block;  
    border-radius: 10px;  
    max-width: 120px;  
    text-align: center;  
    background-color: #527AF2;  
    color: #fff;  
    cursor: pointer;  
    transition: background-color 0.3s;  
    margin-bottom: 10px;  
}
```

```
}
```

```
.btn1:hover{  
    background-color: #666666;  
}
```

```
}
```

```

.inbtn, .delete_btn{
    display: inline-block;
    border-radius: 5px;
    max-width: 100px;
    line-height: 28px;
    text-align: center;
    font-size: 14px;
    cursor: pointer;
    transition: background-color 0.3s; /* 마우스 오버 효과 */
    padding: 7px 15px;
}

.inbtn{
    background-color: #333;
    color: #fff;
}

.delete_btn{
    background-color: red;
    color: #fff;
}

.inbtn:hover, .delete_btn:hover{
    background-color: #555;
}

.frm{
    margin-top: 50px;
}

/*.frm tr > * {*/
/*    margin-right: 10px;*/
/*}*/
.btn_group{
    margin-top: 50px;
}

#ans_btn{
    border-radius: 5px; /* 버튼 크기 */
    text-align: center;
}

.tb3{
    width: 1200px;
    height: 140px;
    margin-bottom: 50px;
    background-color: #eeeeee;
    border-radius: 10px;
}

.tb3 th{
    line-height: 100px;
    text-align: center;
    vertical-align: middle;
}

```

```

.tb3 th:first-child{
    width: 15%;
    font-weight: bold;
    font-size: 18px;
}

.tb3 th:nth-child(2){
    width: 70%;
}

.tb3 th:last-child{
    width: 15%;
}

textarea{
    resize: none;
    padding: 10px;
    height: 80px;
    border: 1px solid #ccc;
    border-radius: 5px;
    vertical-align: middle;
}

#nologin_comment{
    width: 1200px;
    text-align: center;
    padding-top: 60px;
    position: absolute;
    font-weight: bold;
}

</style>
</head>
<body>
<div class="wrap">
    <header class="hd" id="hd">
<%@ include file="../../header.jsp" %>
    </header>
    <div class="contents" id="contents">
        <div class="breadcrumb">
            <p><a href="/">HOME</a>&gt; <a href="">상품</a>&gt; <a href="">상
품구매</a></p>
        </div>
        <section class="page" id="page1">
            <div class="page_wrap">
                <h2 class="page_tit">상품구매</h2>
                <form action="${path
}/AddPaymentPro.do" method="post" onsubmit="return payCheck(this)">
                    <table class="table table-secondary" id="tb1">
                        <tbody>
                            <tr>
                                <td colspan="2">
                                    <c:if test="${!empty pro.imgSrc1}">

```

```

        
        </c:if>
        <hr>
        <c:if test="${!empty pro.imgSrc3}">
            
        </c:if>
        </td>
    </tr>
    <tr>
        <th class="item1">도서 번호</th>
        <td class="item2">${pro.catenno }</td>
    </tr>
    <tr>
        <th>도서명</th>
        <td>
            ${pro.pname }
            <input
type="hidden"name="pname"id="proName"value="${pro.pname }">
            <input
type="hidden"name="from"id="from"value="${from }">
            <c:if test="${from.equals('cart')}">
                <input
type="hidden"name="cartno"id="cartno"value="${cartno }">
            </c:if>
        </td>
    </tr>
    <tr>
        <th>도서 설명</th>
        <td>
            <pre>${pro.pcomment }</pre>
        </td>
    </tr>
    <tr>
        <th>도서 목차</th>
        <td><pre>${pro.plist }</pre></td>
    </tr>
    <tr>
        <th>가격 (부가세 10% 별도)</th>
        <td>
            <!-- 여기서 원가에 *1.1를 하므로 기존 가격보다 더 비쌈 -->
            <fmt:parseNumber
var="price"integerOnly="true"value="${pro.price*1.1 }"/>
            ${price}
            <input
type="hidden"name="sprice"id="sprice"value="${price }">
        </td>
    </tr>
    <tr>

```

```

        <th>구매 수량 (남은 재고 : ${amount})</th>
        <td>
            <input
type="hidden"name="pno"id="pno"value="${pro.no }"/>
            <input
type="number"name="amount"id="amount"class="form-control"min="1"value="1"max
="${amount}"/>
        </td>
    </tr>
</tbody>
</table>
<hr>
<h3>배송 정보</h3>
<table class="table">
    <tbody>
        <tr>
            <th>배송지 주소</th>
            <td><input
type="text"name="address1"id="address1"placeholder="기본 주소"
"class="form-control"required readonly /><br>
            <input
type="text"name="address2"id="address2"placeholder="상세 주소"
"class="form-control"required /><br>
            <input
type="text"name="postcode"id="postcode"style="width:160px;float:left;margin-
right:20px;"placeholder="우편번호"class="form-control"required readonly>
            <button
type="button"id="post_btn"onclick="findAddr()"class="btn
btn-primary"style="margin-bottom:36px;">우편번호 검색</button>
        </td>
    </tr>
    <tr>
        <th>받는 사람 연락처</th>
        <td>
            <input
type="hidden"name="cid"id="cid"value="${cus.id }">
            <input
type="hidden"name="name"id="name"value="${cus.name }">
            <input
type="hidden"name="email"id="email"value="${cus.email }">
            <input
type="hidden"name="tel"id="tel"value="${cus.tel }">
            <input
type="hidden"name="addr"id="addr"value="${cus.address }">
            <input
type="tel"name="custel"id="custel"class="form-control"required>
        </td>
    </tr>
</tbody>
</table>

```



```

<hr>
<h3>결제 정보</h3>
<table class="table">
  <tbody>
    <tr>
      <th>결제 수단</th>
      <td>
        <select
name="pmethod" id="pmethod" class="form-control">
          <option value="신용카드">신용카드</option>
          <option value="체크카드">체크카드</option>
          <option value="계좌이체">계좌이체</option>
        </select>
      </td>
    </tr>
    <tr>
      <th>결제사</th>
      <td>
        <select
name="pcom" id="pcom" class="form-control">
          <option value="선택안함">선택안함</option>
        </select>
        <input
type="hidden" name="pcom2" id="pcom2" value="">
      </td>
    </tr>
    <tr>
      <th>결제 번호</th>
      <td>
        <input
type="text" name="cnum" id="cnum" class="form-control" required>
        <input
type="hidden" name="payAmount" id="payAmount">
        <input
type="hidden" name="payCk" id="payCk" value="no">
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="button" id="pay" value="결제
" class="inbtn">
      </td>
    </tr>
  </tbody>
</table>
<div class="btn-wrap container">
  <c:if test="${!empty sid }">
    <input type="submit" class="inbtn" value="구매">
  </c:if>
  <a href="${path }/ProList.do" class="inbtn">제품 목록

```

```

</a>
<p id="paymentResult" style="color:red; font-size:
20px;"></p>
</div>
</form>
<script>
$(document).ready(function(){
    var cardArr1 = ["현대카드", "농협카드", "BC카드", "KB카드"];
    var cardArr2 = ["하나카드", "농협카드", "BC카드"];
    var bankArr = ["카카오뱅크", "농협은행", "신한은행", "기업은
행", "국민은행"];
    $("#pmethod").change(function(){
        var th = $(this).val();
        if(th === "신용카드"){
            for(var i=0; i<cardArr1.length; i++) {
                $("#pcom").append("<option value='"+ cardArr1[i] + "'>" + cardArr1[i] +
"</option>");
            }
        } else if(th === "체크카드"){
            for(var i=0; i<cardArr2.length; i++) {
                $("#pcom").append("<option value='"+ cardArr2[i] + "'>" + cardArr2[i] + "</option>");
            }
        } else {
            for(var i=0; i<bankArr.length; i++) {
                $("#pcom").append("<option value='"+ bankArr[i] + "'>" + bankArr[i] + "</option>");
            }
        }
        $("#pcom").change(function(){
            $("#pcom2").val($(this).val());
        });
    });
</script>
<script>
    //주소 연동 API
    function findAddr() {
        new daum.Postcode({
            oncomplete: function(data) {
                console.log(data);
                var roadAddr = data.roadAddress;
                var jibunAddr = data.jibunAddress;
                document.getElementById("postcode").value = data.zonecode;
                if(roadAddr !== '') {
                    document.getElementById("address1").value = roadAddr;
                } else if(jibunAddr !== '') {
                    document.getElementById("address1").value = jibunAddr;
                }
                document.getElementById("address2").focus();
            }
        }).open();
    }

```

```

}
</script>
<script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
<script>
    //결제모듈 API 연동
$(document).ready(function(){
    var totalPay=0;
    var proName;
$("#pay").click(function(){
    var email = $("#name").val();
    var cname = $("#email").val();
    var tel = $("#tel").val();
    var addr = $("#addr").val();
    var postcode = $("#postcode").val();
proName = $("#proName").val();
    if($("#amount").val()!="") {
totalPay = parseInt($("#sprice").val()) * parseInt($("#amount").val());
    } else{
alert("구매할 수량을 입력하지 않으셨습니다.");
$("#amount").focus();
    }
    return;
}
alert("결제할 금액 : "+totalPay);
    //상품명_현재시간
    var date = new Date();
    var date = d.getFullYear()+''+(d.getMonth()+1)+''+d.getDate()+''+d.getHours()+''+d.getMinutes()+''+d.getSeconds();
    var IMP = window.IMP; // 생략가능
IMP.init('imp85122075'); // 결제 API를 사용하기 위한 코드 입력!
IMP.request_pay({
    //결제 요청
pg: "T5102001",
merchant_uid : '상품명_'+ date, //상점 거래 ID
name : proName, // 결제 명
amount : 10, // 결제금액
buyer_email : email, // 구매자 email
buyer_name : cname, // 구매자 이름
buyer_tel : tel, // 구매자 전화번호
buyer_addr : addr, // 구매자 주소
buyer_postcode : postcode // 구매자 우편번호
}, function(rsp){
    if(rsp.success){
console.log(rsp);
    var msg = '결제가 완료 되었습니다.';
    var r1 = '고유 아이디 : '+rsp.imp_uid;
    var r2 = '상점 거래 아이디 : '+rsp.merchant_uid;
    var r3 = '결제 금액 : '+rsp.paid amount;

```

```

        var r4 = '카드 승인 번호 : '+rsp.apply_num;
        // 실제 결제 창
        // $("#payCk").val("yes");
        // $("#payAmount").val(rsp.paid_amount);
        // $("#pmethod").val(rsp.pay_method);
        // $("#pcom").val(rsp.pg_provider);
        // $("#cnum").val(rsp.apply_num);
        // alert(msg);
    }
    // $("#paymentResult").html(r1+"<br>" + r2 + "<br>" + r3 + "<br>" + r4);
} else{
    // $("#paymentResult").html('결제실패<br>' +
    에러내용 : ' +rsp.error_msg);
}
// 테스트용이므로 실패시에도 그냥 통과시킴
$("#payCk").val("yes");
$("#payAmount").val(totalPay);
$("#pmethod").val("신용카드");
$("#pcom").val("삼성카드");
$("#cnum").val("123-1234-1234-1278");
$("#paymentResult").text("결제 완료 : " + totalPay);
});
});
});

</script>
<script>
    function payCheck(f){
        var payCk = f.payCk.value;
        console.log(payCk);
        if(payCk != "yes"){
            alert("아직 결제 전 입니다.");
            return false;
        }
    }
</script>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="../../footer.jsp" %>
</footer>
</div>
</body>
</html>

```

결제목록

연번	결제정보	결제상품	작업
1	구매 수량 : 1 결제 방법 : 신용카드 결제 대행 : 농협카드 결제 번호 : 123-1234-1234-1278 결제 금액 : 26400	고등국어1	출고처리중 반품요청

Search:

Previous1Next

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>결제 목록</title>
<%@ include file="../head.jsp"%>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css"rel="styles
heet">
    <!-- 플러그인 연결-->
    <!-- jquery -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- datatables -->
    <link
rel="stylesheet"href="https://cdn.datatables.net/1.13.6/css/jquery.dataTable
s.css">
    <script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet"href="{path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet"href="{path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet"href="{path}/css/common.css">
    <link rel="stylesheet"href="{path}/css/hd.css">
    <link rel="stylesheet"href="{path}/css/ft.css">
<style>
    /* 본문 영역 스타일 */
```

```
.contents{
    clear: both;
    min-height: 200vh;
    background-image: url("${path}/img/login.jpg");
    background-repeat: no-repeat;
    background-position: center -250px;
}

.contents::after{
    content: "";
    clear: both;
    display: block;
    width: 100%;
}

.page{
    clear: both;
    width: 100vw;
    height: 100vh;
    position: relative;
}

.page::after{
    content: "";
    display: block;
    width: 100%;
    clear: both;
}

.page_wrap{
    clear: both;
    width: 1200px;
    height: auto;
    margin: 0 auto;
}

.page_tit{
    font-size: 48px;
    text-align: center;
    /*padding-top: 0.7em;*/
    color: #fff;
    padding-bottom: 1.3em;
}

.breadcrumb{
    clear: both;
    width: 1200px;
    margin: 0 auto;
    text-align: right;
    color: #fff;
    padding-top: 28px;
    padding-bottom: 28px;
}

.breadcrumb a{
    color: #fff;
}
```

```

/* 테이블 스타일 */
.tb1{
    width: 80%;
    margin: 50px auto;
    font-size: 20px;
    border-collapse: collapse;
}

.tb1 th{
    background-color: #527AF2;
    color: #fff;
    padding: 16px;
    border: 1px solid #527AF2;
}

.tb1 th:first-child{
    width: 40px;
}

#myTable td{
    max-height: 200px;
    overflow: hidden; /* Overflow 처리 (필요한 경우) */
    vertical-align: top; /* 상단 정렬로 변경 */
}

#myTable td > div{
    text-align: center;
}

.img_tit{
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100%;
}

#myTable tbody tr:not(:last-child) td{
    border-bottom: 1px solid #333;
}

.img_tit img{
    max-height: 180px;
}

.tb1 .item1{
    width: 10%;
}

.tb1 .item2{
    width: 50%;
    text-align: left;
}

.tb1 .item3{
    width: 25%;
    text-align: right;
}

/* 기타 버튼 스타일 */
.btn1{
    border-radius: 10px;
}

```

```

        padding-left: 10px;
        padding-right: 10px;
        max-width: 100px;
        text-align: center;
        background-color: #527AF2;
        color: #fff;
        cursor: pointer;
        transition: background-color 0.3s;
    }

    .btn1:hover{
        background-color: #666666;
    }

    .inbtn{
        display: block;
        border-radius: 10px;
        min-width: 60px;
        padding-left: 24px;
        padding-right: 24px;
        text-align: center;
        line-height: 38px;
        background-color: #527AF2;
        color: #fff;
        font-size: 18px;
        float: right;
        cursor: pointer;
        transition: background-color 0.3s;
    }

    .inbtn:hover{
        background-color: #666666;
    }

    .btn_group{
        margin-top: -38px;
        z-index: 1000;
        position: relative;
    }

    .btn_group p{
        float: right;
    }
}

</style>
</head>
<body>
<div class="container-fluid">
    <header class="hd" id="hd">
<%@ include file="../header.jsp" %>
    </header>
    <div class="contents" id="contents">
        <div class="breadcrumb">
            <p><a href="/">HOME</a>&gt; <a href="/">결제목록</a></p>
        </div>
        <section class="page" id="page1">

```



```

<div class="page_wrap">
  <h2 class="page_tit">결제목록</h2>
  <table class="tbl1" id="myTable">
    <thead>
      <tr>
        <th class="item1">연번</th>
        <th class="item2">결제정보</th>
        <th class="item3">결제상품</th>
        <th class="item4">작업</th>
      </tr>
    </thead>
    <c:forEach var="pay" items="${payList }" varStatus="status">
      <tr>
        <td class="item1">${status.count }</td>
        <td class="item2">
          <p>구매 수량 : ${pay.amount }</p>
          <p>결제 방법 : ${pay.pmethod }</p>
          <p>결제 대행 : ${pay.pcom }</p>
          <p>결제 번호 : ${pay.cnum }</p>
          <p>결제 금액 : ${pay.payprice }</p>
        </td>
        <td class="item3">
          <span title="${pay.pno }">${pay.pname }</span>
        </td>
        <td class="item4">
          <c:if test="${pay.pstate==0}">
            <span>출고처리중</span>
            <a href="${path
}/ReturnPayment.do?sno=${pay.sno }" class="inbtn">반품요청</a>
          </c:if>
          <c:if test="${pay.pstate==1}">
            <span>배송중</span>
            <a href="${path
}/CusDelivery.do?sno=${pay.sno }" class="inbtn">배송조회</a>
          </c:if>
          <c:if test="${pay.pstate==2}">
            <span>배송완료</span>
            <a href="${path
}/ReturnPayment.do?sno=${pay.sno }" class="inbtn">환불요청</a>
            <a href="${path
}/CusDelivery.do?sno=${pay.sno }" class="inbtn">배송조회</a>
            <a href="${path
}/BuyPayment.do?sno=${pay.sno }" class="inbtn">구매확정</a>
          </c:if>
          <c:if test="${pay.pstate==3}">
            <span>구매 확정</span>
          </c:if>
        </td>
      </tr>
    </c:forEach>
  </div>

```

```

        </table>
    </div>
    <script>
$(document).ready( function() {
$('#myTable').DataTable({
    pageLength : 5,
    order: [[0, 'desc']], // 0번째 컬럼을 기준으로 내림차순 정렬
    info: false,
    dom: 't<f>p',
    language: {
        emptyTable: '결제 내역이 없습니다.'
    }
});
} );
$(document).ready(function() {
$('.dataTables_paginate').css({
    'textAlign': 'left',
    'float': 'none',
    'margin-top': '10px',
});
$('.dataTables_filter').css({
    'float': 'left',
    'margin-top': '14px',
    'margin-right': '280px'
});
$('#myTable_paginate').css({
    'margin-right': '100px'
});
});
    </script>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="../footer.jsp" %>
</footer>
</div>
</div>
</body>
</html>

```

출고처리

Show 10 entries

배송번호	구매자	주소	비고
1	admin	경기 성남시 분당구 대왕판교로 477 102호 13480	배송 상세

Search:

Previous 1 Next

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
  <c:set var="path" value="%=request.getContextPath() %=" />
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>배송 정보 작성</title>
<%@ include file="/head.jsp" %>
  <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
  <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
  <!-- 플러그인 연결-->
  <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
  <!-- 스타일 초기화 -->
  <!-- 스타일 초기화 -->
  <link rel="stylesheet" href="${path}/css/reset.css">
  <!-- 웹 폰트 -->
  <link rel="stylesheet" href="${path}/css/font.css">
  <!-- css 모듈화 -->
  <link rel="stylesheet" href="${path}/css/common.css">
  <link rel="stylesheet" href="${path}/css/hd.css">
  <link rel="stylesheet" href="${path}/css/ft.css">
  <style>
    /* 본문 영역 스타일 */
    .contentsclear:both; min-height: 100vh; background-image:
url("${path}/img/login.jpg");
    background-repeat: no-repeat; background-position: center -250px;
  }
```

```

.contents::aftercontent:""; clear:both; display:block; width:100%; }
.pageclear:both; width: 100vw; height: 200vh; position:relative; }
.page::aftercontent:""; display:block; width: 100%; clear:both; }
.page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
.page_titfont-size:48px; text-align: center; color:#fff;
padding-bottom: 1.3em; }
.breadcrumbclear:both;
width:1200px; margin: 0 auto; text-align: right; color:#fff;
padding-top: 28px; padding-bottom: 28px; }
.breadcrumbb acolor:#fff; }
.tb1width:600px; margin:0 auto; font-size: 24px;}
.tb1 thwidth: 200px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
.tb1 tdwidth: 400px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
.tb1 td:last-childline-height: 48px; padding-top:24px;
padding-bottom:24px; }

```

```

.inbtndisplay:block;
border-radius:10px;
min-width:120px;
padding-left: 24px;
padding-right: 24px;
text-align: center;
line-height: 38px;
background-color: #333;
color:#fff;
font-size: 18px;
cursor: pointer; }
.inbtn:first-childfloat:left; }
.inbtn:last-childfloat:right; }
.inbtn:hover{
background-color: #666666;
}

```

```

.indata{
display:inline-block;
width: 590px;
height: 32px;
line-height: 32px;
text-indent:10px;
font-size:16px;
border: 1px solid #ccc;
border-radius: 5px;
padding: 5px 15px;
}

```

```

        textarea{
            resize: none;
            width: 600px;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 5px;
            font-size: 16px;
        }
    </style>
</head>
<body>
<div class="wrap">
    <header class="hd" id="hd">
<%@ include file="/header.jsp" %>
    </header>
    <div class="contents" id="contents">
        <div class="breadcrumb">
            <p><a href="/">HOME</a>&gt; <a href="/">배송정보</a>
&gt; <a href="/">배송정보</a></p>
        </div>
        <section class="page" id="page1">
            <div class="page_wrap">
                <h2 class="page_tit">배송정보</h2>
                <form
action="${path}/UpdateDeliveryPostPro.do" class="frm" method="post">
                    <table class="tb1">
                        <tbody>
                            <tr>
                                <th>배송 분류</th>
                                <td colspan="2">
                                    <select
name="pstate" id="pstate" class="indata" autofocus required>
                                        <option value="0"${deli.pstate eq
"0"? 'selected': ''}>출고</option>
                                        <option value="1"${deli.pstate eq
"1"? 'selected': ''}>배송중</option>
                                        <option value="2"${deli.pstate eq
"2"? 'selected': ''}>배송완료</option>
                                        <option value="3"${deli.pstate eq
"3"? 'selected': ''}>구매완료</option>
                                    </select>
                                </td>
                                <input
type="hidden" name="dno" id="dno" value="${deli.dno}" readonly>
                                </tr>
                                <tr>
                                    <th>구매자</th>
                                    <td colspan="2"><input
type="text" name="cid" id="cid" class="indata" value="${deli.cid}" readonly>
                                </td>
                                </tr>

```

```

        <tr>
            <th>주소</th>
            <td colspan="2"><input
type="text"name="daddr"id="daddr"class="indata"value="${deli.daddr}"readonly
></td>
        </tr>
        <tr>
            <th>배송회사</th>
            <td colspan="2"><input
type="text"name="pcom"id="pcom"class="indata"value="${deli.pcom}"required></
td>
        </tr>
        <tr>
            <th>기사번호</th>
            <td colspan="2"><input
type="text"name="ptel"id="ptel"class="indata"value="${deli.ptel}"required></
td>
        </tr>
        <tr>
            <th>운송번호</th>
            <td colspan="2"><input
type="text"name="bcode"id="bcode"class="indata"value="${deli.bcode}"required
></td>
        </tr>
        <tr>
            <th>배송날짜</th>
            <td colspan="2"><input
type="text"name="rdate"id="rdate"class="indata"value="${deli.rdate}"required
maxlength="13"></td>
        </tr>
        <tr>
            <td colspan="2">
                <c:if test="${deli.pstate ne '3'}">
                    <input type="submit"value="상품등록
"class="inbtn">
                </c:if>
                <input type="reset"value="취소
"class="inbtn"onclick="window.history.back();">
                <a href="${path}/ProList.do"class="inbtn">상품
목록</a>
            </td>
        </tr>
    </tbody>
</table>
</form>
</div>
</section>
</div>
<footer class="ft"id="ft">
<%@ include file="/footer.jsp" %>

```

```
</footer>
</div>
</body>
</html>
```

배송조회

HOME > 결제목록

결제목록

연번	결제정보	결제상품	작업
1	구매 수량 : 1 결제 방법 : 신용카드 결제 대행 : 농협카드 결제 번호 : 123-1234-1234-1278 결제 금액 : 26400	고등국어1	배송완료
			환불요청
			배송조회
			구매확정

Search:

Previous 1 Next

배송정보

배송번호	1
구매번호	1
구매상품	고등국어1
구매수량	1
구매자	admin
주소	경기 성남시 분당구 대왕판교로 477 102호 13480
전화번호	010-1111-1111
운송회사	대한통운
운송번호	gzjrXBll7V4578
기사 전화번호	010-5678-5555
구매번호	1
배송정보	배송완료
결제날짜	2023-09-04 13:05:27
출고날짜	23-08-31

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
  <c:set var="path" value="<%=request.getContextPath() %>" />
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>배송정보</title>
<%@ include file="/head.jsp" %>
  <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
  <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
  <!-- 플러그인 연결 -->
  <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
  <!-- 스타일 초기화 -->
  <!-- 스타일 초기화 -->
  <link rel="stylesheet" href="${path}/css/reset.css">
  <!-- 웹 폰트 -->
  <link rel="stylesheet" href="${path}/css/font.css">
  <!-- css 모듈화 -->
  <link rel="stylesheet" href="${path}/css/common.css">
  <link rel="stylesheet" href="${path}/css/hd.css">
  <link rel="stylesheet" href="${path}/css/ft.css">
<style>
  /* 본문 영역 스타일 */
    .contentsclear:both; min-height: 100vh; background-image:
url("${path}/img/login.jpg");
    background-repeat: no-repeat; background-position: center -250px;
}

    .contents::aftercontent:""; clear:both; display:block; width:100%; }
    .pageclear:both; width: 100vw; height: 200vh; position:relative; }
    .page::aftercontent:""; display:block; width: 100%; clear:both; }
    .page_wrapclear:both; width: 1200px; height: auto; margin:0 auto; }
    .page_titfont-size:48px; text-align: center; color:#fff;
padding-bottom: 1.3em; }
    .breadcrumbclear:both;
width:1200px; margin: 0 auto; text-align: right; color:#fff;
padding-top: 28px; padding-bottom: 28px; }
    .breadcrumb acolor:#fff; }
    .tblwidth:600px; margin:0 auto; font-size: 24px;}
    .tbl thwidth: 200px; line-height: 32px; padding-top:16px;
padding-bottom:16px;
border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
    .tbl tdwidth: 400px; line-height: 32px; padding-top:16px;
padding-bottom:16px;

```



```

        border-bottom: 1px solid #333; border-top: 1px solid #333;
box-sizing: border-box; text-align: center;}
        .tbl td:last-childline-height: 48px; padding-top:24px;
padding-bottom:24px; }

```

```

.inbtndisplay:block;
border-radius:10px;
min-width:120px;
padding-left: 24px;
padding-right: 24px;
text-align: center;
line-height: 38px;
background-color: #333;
color:#fff;
font-size: 18px;
cursor: pointer; }
.inbtn:first-childfloat:left; }
.inbtn:last-childfloat:right; }
.inbtn:hover{
background-color: #666666;
}

```

```

.indata{
display:inline-block;
width: 590px;
height: 32px;
line-height: 32px;
text-indent:10px;
font-size:16px;
border: 1px solid #ccc;
border-radius: 5px;
padding: 5px 15px;
}

```

```

textarea{
resize: none;
width: 600px;
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px;
font-size: 16px;
}

```

```

</style>
</head>
<body>
<div class="wrap">
    <header class="hd" id="hd">
<%@ include file="/header.jsp" %>
    </header>
    <div class="contents" id="contents">
        <div class="breadcrumb">

```

```

        <p><a href="/">HOME</a>&gt; <a href="/">배송정보</a>
&gt; <a href="/">배송정보</a></p>
    </div>
    <section class="page" id="page1">
        <div class="page_wrap">
            <h2 class="page_tit">배송정보</h2>
            <table class="tb1">
                <tbody>
                    <tr>
                        <th>배송번호</th>
                        <td colspan="2">${deli.dno}</td>
                    </tr>
                    <tr>
                        <th>구매번호</th>
                        <td colspan="2">${deli.sno}</td>
                    </tr>
                    <tr>
                        <th>구매상품</th>
                        <td colspan="2">${deli.pname}</td>
                    </tr>
                    <tr>
                        <th>구매수량</th>
                        <td colspan="2">${deli.amount}</td>
                    </tr>
                    <tr>
                        <th>구매자</th>
                        <td colspan="2">${deli.cid}</td>
                    </tr>
                    <tr>
                        <th>주소</th>
                        <td colspan="2">${deli.daddr}</td>
                    </tr>
                    <tr>
                        <th>전화번호</th>
                        <td colspan="2">${deli.custel}</td>
                    </tr>
                    <tr>
                        <th>운송회사</th>
                        <td colspan="2">${deli.pcom}</td>
                    </tr>
                    <tr>
                        <th>운송번호</th>
                        <td colspan="2">${deli.bcode}</td>
                    </tr>
                    <tr>
                        <th>기사 전화번호</th>
                        <td colspan="2">${deli.ptel}</td>
                    </tr>
                    <tr>
                        <th>구매번호</th>

```

```

        <td colspan="2">${deli.sno}</td>
    </tr>
    <tr>
        <th>배송정보</th>
        <c:if test="${deli.pstate == 0}">
            <td colspan="2">출고처리중</td>
        </c:if>
        <c:if test="${deli.pstate == 1}">
            <td colspan="2">배송중</td>
        </c:if>
        <c:if test="${deli.pstate == 2}">
            <td colspan="2">배송완료</td>
        </c:if>
        <c:if test="${deli.pstate == 3}">
            <td colspan="2">구매완료</td>
        </c:if>
    </tr>
    <tr>
        <th>결제날짜</th>
        <td colspan="2">${deli.sdate}</td>
    </tr>
    <tr>
        <th>출고날짜</th>
        <td colspan="2">${deli.rdate}</td>
    </tr>
</tbody>
</table>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="/footer.jsp" %>
</footer>
</div>
</body>
</html>

```

관리자 페이지

Show 10 entries

번호	이름	판매수익	판매량
1	고등국어1	26400	1

Search:

Previous 1 Next

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="path" value="<%=request.getContextPath() %>"/>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>판매량페이지</title>
<%@ include file="/head.jsp" %>
    <!-- 스타일 초기화 : reset.css 또는 normalize.css -->
    <link
href="https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css" rel="styles
heet">
    <!-- 플러그인 연결-->
    <!-- jquery -->
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"></script>
    <!-- datatables -->
    <link
rel="stylesheet" href="https://cdn.datatables.net/1.13.6/css/jquery.dataTables
s.css">
    <script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTables.js"></script>
    <!-- 스타일 초기화 -->
    <link rel="stylesheet" href="{path}/css/reset.css">
    <!-- 웹 폰트 -->
    <link rel="stylesheet" href="{path}/css/font.css">
    <!-- css 모듈화 -->
    <link rel="stylesheet" href="{path}/css/common.css">
    <link rel="stylesheet" href="{path}/css/hd.css">
    <link rel="stylesheet" href="{path}/css/ft.css">
```

```

<style>
  /* 본문 영역 스타일 */
  .contents{
    clear: both;
    min-height: 100vh;
    background-image: url("${path}/img/login.jpg");
    background-repeat: no-repeat;
    background-position: center -250px;
  }

  .contents::after{
    content: "";
    clear: both;
    display: block;
    width: 100%;
  }

  .page{
    clear: both;
    width: 100vw;
    height: 100vh;
    position: relative;
  }

  .page::after{
    content: "";
    display: block;
    width: 100%;
    clear: both;
  }

  .page_wrap{
    clear: both;
    width: 1200px;
    height: auto;
    margin: 0 auto;
  }

  .page_tit{
    font-size: 48px;
    text-align: center;
    /*padding-top: 0.7em;*/
    color: #fff;
    padding-bottom: 1.3em;
  }

  .breadcrumb{
    clear: both;
    width: 1200px;
    margin: 0 auto;
    text-align: right;
    color: #fff;
    padding-top: 28px;
    padding-bottom: 28px;
  }

  .breadcrumb a{

```

```

        color: #fff;
    }

    /* 테이블 스타일 */
    .tb1{
        width: 80%;
        margin: 50px auto;
        font-size: 20px;
        border-collapse: collapse;
    }

    .tb1 th{
        background-color: #527AF2;
        color: #fff;
        padding: 16px;
        border: 1px solid #527AF2;
    }

    .tb1 td{
        padding: 12px 16px;
        border: 1px solid #ddd;
        text-align: center;
        line-height: 24px;
    }

    .tb1 th:first-child{
        width: 40px;
    }

    .tb1 tbody{
    }

    .tb1 .item1{
        width: 10%;
    }

    .tb1 .item2{
        width: 20%;
    }

    /* 기타 버튼 스타일 */
    .inbtn{
        display: block;
        border-radius: 10px;
        min-width: 100px;
        text-align: center;
        background-color: #527AF2;
        color: #fff;
        float: right;
        cursor: pointer;
        transition: background-color 0.3s;
    }

    .inbtn:hover{
        background-color: #666666;
    }

    .btn_group{
        margin-top: -38px;
        z-index: 1000;
    }

```

```

        position: relative;
    }
    .btn_group p{
        float: right;
    }
</style>
</head>
<body>
<div class="wrap">
    <header class="hd" id="hd">
<%@ include file="/header.jsp" %>
    </header>
    <div class="contents" id="contents">
        <div class="breadcrumb">
            <p><a href="/">HOME</a>&gt; <a href="/admin/baordList.jsp">관리자
페이지</a></p>
        </div>
        <section class="page" id="page1">
            <div class="page_wrap">
                <h2 class="page_tit">관리자 페이지</h2>
                <table class="tb1" id="myTable">
                    <thead>
                        <tr>
                            <th class="item1" style="text-align: center">번호
                        </th>
                            <th class="item3" style="text-align: center">이름
                        </th>
                            <th class="item2" style="text-align: center">판매수
                        </th>
                            <th class="item3" style="text-align: center">판매량
                        </th>
                        </tr>
                    </thead>
                    <tbody>
                        <c:forEach var="cus" items="${salesList}" varStatus="status">
                            <tr>
                                <td class="item1">${status.count}</td>
                                <td class="item2">${cus.pname}</td>
                                <td class="item2">${cus.money}</td>
                                <td class="item2">${cus.cnt}</td>
                            </tr>
                        </c:forEach>
                    </tbody>
                </table>
                <script>
$(document).ready( function() {
$('#myTable').DataTable({
order: [[3, 'desc']], // 0번째 컬럼을 기준으로 내림차순 정렬
info: false,

```

```

dom: 'lt<f>p',
language: {
emptyTable: '판매 내역이 없습니다.'
}
});
} );
$(document).ready(function() {
$('.dataTables_paginate').css({
'textAlign': 'left',
'float': 'none',
'margin-top': '10px',
});
$('.dataTables_filter').css({
'float': 'left',
'margin-top': '14px',
'margin-right': '280px'
});
$('#myTable_paginate').css({
'margin-right': '100px'
});

});

</script>
</div>
</section>
</div>
<footer class="ft" id="ft">
<%@ include file="/footer.jsp" %>
</footer>
</div>
</body>
</html>

```


해당 능력단위 요소	배치 프로그램 구현하기	(20점)
수행5. 작성 내용		

1. 회원 SQL, DAO

DAO

```
package shop.model;
```

```
import shop.dto.Custom;
import shop.util.AES256;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
public class CustomDAO {
    static Connection conn = null;
    static PreparedStatement pstmt = null;
    static ResultSet rs = null;
    static String key = "%02x";
    public List<Custom> getCustomList() {
        List<Custom> cusList = new ArrayList<>();
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        try{
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_SELECT_ALL);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                Custom custom = new Custom();
                custom.setId(rs.getString("id"));
                custom.setPw(rs.getString("pw"));
                custom.setName(rs.getString("name"));
                custom.setPoint(rs.getInt("point"));
                custom.setGrade(rs.getString("grade"));
                custom.setTel(rs.getString("tel"));
                custom.setEmail(rs.getString("email"));
                custom.setBirth(rs.getString("birth"));
                custom.setRegdate(rs.getString("regdate"));
                cusList.add(custom);
            }
        } catch(SQLException e) {
            System.out.println("sql 에러");
        } finally{
            con.close(rs, pstmt, conn);
        }
        return cusList;
    }
}
```

```

}

    public Custom getCustom(String id) {
        Custom custom = new Custom();
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        String qpww = "";
        try{
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_SELECT_ONE);
            pstmt.setString(1, id);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                custom.setId(rs.getString("id"));
                try{
                    qpww = AES256.decryptAES256(rs.getString("pw"), key);
                } catch (Exception e) {
                    throw new RuntimeException(e);
                }
            }
            custom.setPw(qpww);
            custom.setName(rs.getString("name"));
            custom.setPoint(rs.getInt("point"));
            custom.setGrade(rs.getString("grade"));
            custom.setTel(rs.getString("tel"));
            custom.setEmail(rs.getString("email"));
            custom.setBirth(rs.getString("birth"));
            custom.setRegdate(rs.getString("regdate"));
            custom.setAddress(rs.getString("address"));
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally{
            con.close(rs, pstmt, conn);
        }

        return custom;
    }

    public boolean login(String id, String pw) {
        boolean pass = false;
        DBConnect con = new MariaDBCon();
        String qpww = "";
        try{
            conn = con.connect();
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_SELECT_LOG);
            pstmt.setString(1, id);
            rs = pstmt.executeQuery();
            if(rs.next()){
                try{
                    qpww = AES256.decryptAES256(rs.getString("pw"), key);
                } catch (Exception e) {
                    throw new RuntimeException(e);
                }
            }

            if(pw.equals(qpww)){

```

```

pass = true;
} else{
pass = false;
}
} else{
pass = false;
}
} catch(SQLException e) {
    throw new RuntimeException(e);
} finally{
con.close(rs, pstmt, conn);
}

    return pass;
}

    public boolean join(String id, String pw, String name, String tel,
String email, String birth, String address) {
        boolean pass = false;
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        try{
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_INSERT);
            pstmt.setString(1, id);
            pstmt.setString(2, pw);
            pstmt.setString(3, name);
            pstmt.setString(4, tel);
            pstmt.setString(5, email);
            pstmt.setString(6, birth);
            pstmt.setString(7, address);
            int cnt = pstmt.executeUpdate();
            if(cnt > 0) {
                pass = true;
            }
        } catch(SQLException e) {
            System.out.println("sql 에러");
        } finally{
            con.close(pstmt, conn);
        }

        return pass;
    }

    public boolean idCheck(String id) {
        Custom custom = new Custom();
        DBConnect con = new MariaDBCon();
        boolean pass = false;
        conn = con.connect();
        try{
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_SELECT_ONE);
            pstmt.setString(1, id);
            rs = pstmt.executeQuery();
            if(!rs.next()) {
                pass = true;
            }
        }
    }

```

```

}
} catch(SQLException e) {
    throw new RuntimeException(e);
} finally{
con.close(rs, pstmt, conn);
}

    return pass;
}

    public int deleteCustom(String id){
        int cnt = 0;
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        if(conn != null){
            System.out.println("SQL 연결 성공");
        }
        try{
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_DELETE);
            pstmt.setString(1, id);
            cnt = pstmt.executeUpdate();
        } catch(SQLException e) {
            throw new RuntimeException(e);
        } finally{
            con.close(pstmt, conn);
        }

        return cnt;
    }

    public int modifyCustom(Custom cus) {
        int cnt = 0;
        DBConnect con = new MariaDBCon();
        String qpww = "";
        try{
            conn = con.connect();
            pstmt = conn.prepareStatement(DBConnect.CUSTOM_UPDATE);
            try{
                qpww = AES256.encryptAES256(cus.getPw(), key);
            } catch(Exception e) {
                throw new RuntimeException(e);
            }
            System.out.println("암호화 : "+qpww);
            pstmt.setString(1, qpww);
            pstmt.setString(2, cus.getTel());
            pstmt.setString(3, cus.getEmail());
            pstmt.setString(4, cus.getAddress());
            pstmt.setString(5, cus.getBirth());
            pstmt.setString(6, cus.getId());
            cnt = pstmt.executeUpdate();
        } catch(SQLException e) {
            throw new RuntimeException(e);
        } finally{
            con.close(pstmt, conn);
        }
    }

```

```

}
    return cnt;
}
}

```

회원 SQL

```

final static String CUSTOM_SELECT_ALL = "select * from custom order by regdate desc";
final static String CUSTOM_SELECT_ONE = "select * from custom where id=?";
final static String CUSTOM_INSERT = "insert into custom values(?, ?, ?, default, default, ?, ?, ?, default, ?)";
final static String CUSTOM_UPDATE = "update custom set pw=?, tel=?, email=?, address=?, birth=? where id=?";
final static String CUSTOM_DELETE = "delete from custom where id=?";
final static String CUSTOM_SELECT_LOG = "select * from custom where id=?";

```

게시판 DAO

```

package shop.model;
import shop.dto.Notice;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
public class NoticeDAO {
    static Connection conn = null;
    static PreparedStatement pstmt = null;
    static ResultSet rs = null;
    public List<Notice> getNoticeList(){
        List<Notice> notiList = new ArrayList<>();
        DBConnect con = new MariaDBCon();
        SimpleDateFormat ymd = new SimpleDateFormat("yy-MM-dd");
        try{
            conn = con.connect();
            pstmt = conn.prepareStatement(DBConnect.NOTICE_SELECT_ALL);
            rs = pstmt.executeQuery();
            while(rs.next()){
                Notice noti = new Notice();
                noti.setNo(rs.getInt("no"));
                noti.setTitle(rs.getString("title"));
                noti.setContent(rs.getString("content"));
                Date d = ymd.parse(rs.getString("resdate")); //날짜데이터로 변경
                String date = ymd.format(d);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return notiList;
    }
}

```

```

noti.setResdate(date);
noti.setVisited(rs.getInt("visited"));
notiList.add(noti);
}
} catch(SQLException) {
    throw new RuntimeException(e);
} catch(ParseException) {
    throw new RuntimeException(e);
} finally{
con.close(rs, pstmt, conn);
}

    return notiList;
}

    public Notice getNotice(int no){
        Notice noti = new Notice();
        DBConnect con = new MariaDBCon();
        SimpleDateFormat ymd = new SimpleDateFormat("yy-MM-dd");
        conn = con.connect();
        if(conn != null){
            System.out.println("PostgreSQL 연결 성공");
        }

        try{
            pstmt = conn.prepareStatement(DBConnect.NOTICE_SELECT_ONE);
            pstmt.setInt(1, no);
            rs = pstmt.executeQuery();
            if(rs.next()){
                noti.setNo(rs.getInt("no"));
                noti.setTitle(rs.getString("title"));
                noti.setContent(rs.getString("content"));
                Date d = ymd.parse(rs.getString("resdate")); //날짜데이터로 변경
                String date = ymd.format(d);
                noti.setResdate(date);
                noti.setVisited(rs.getInt("visited"));
            }
        } catch(SQLException) {
            throw new RuntimeException(e);
        } catch(ParseException) {
            throw new RuntimeException(e);
        } finally{
            con.close(rs, pstmt, conn);
        }
        updateCnt(no);
        return noti;
    }

    // 조회수 갱신
    public void updateCnt(int no) {
        Notice noti = new Notice();
        DBConnect con = new MariaDBCon();

```

```

conn = con.connect();
    try{
pstmt = conn.prepareStatement(DBConnect.NOTICE_UPDATE_CNT);
pstmt.setInt(1, no);
        intcnt = pstmt.executeUpdate();
    } catch(SQLException e) {
        throw new RuntimeException(e);
    } finally{
con.close(pstmt, conn);
    }
}

```

```

    public int addNotice(Notice noti){
        intcnt = 0;
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{
pstmt = conn.prepareStatement(DBConnect.NOTICE_INSERT);
pstmt.setString(1, noti.getTitle());
pstmt.setString(2, noti.getContent());
cnt = pstmt.executeUpdate();
    } catch(SQLException e) {
        throw new RuntimeException(e);
    } finally{
con.close(pstmt, conn);
    }

        return cnt;
    }
}

```

```

    public int updateNotice(Notice noti){
        intcnt = 0;
        DBConnect con = new MariaDBCon();
conn = con.connect();
        if(conn!=null){
System.out.println("Maria updateNotice 연결 성공");
        }

        try{
pstmt = conn.prepareStatement(DBConnect.NOTICE_UPDATE);
pstmt.setString(1, noti.getTitle());
pstmt.setString(2, noti.getContent());
pstmt.setInt(3, noti.getNo());
cnt = pstmt.executeUpdate();
    } catch(SQLException e) {
        throw new RuntimeException(e);
    } finally{
con.close(pstmt, conn);
    }

        return cnt;
    }
}

    public int deleteNotice(int no){
        intcnt = 0;

```

```

        DBConnect con = new MariaDBCon();
conn = con.connect();
        if(conn!=null){
System.out.println("Maria deleteNotice 연결 성공");
}

        try{
pstmt = conn.prepareStatement(DBConnect.NOTICE_DELETE);
pstmt.setInt(1, no);
cnt = pstmt.executeUpdate();
} catch(SQLException) {
        throw new RuntimeException(e);
} finally{
con.close(pstmt, conn);
}

        return cnt;
}
}

```

게시판 SQL

```

final static String NOTICE_SELECT_ALL = "select * from notice";
        final static String NOTICE_SELECT_ONE = "select * from notice where no=?";
        final static String NOTICE_INSERT = "insert into notice values
(default,?,?,?,default,default)";
        final static String NOTICE_UPDATE = "update notice set title=?, content=?
where no=?";
        final static String NOTICE_UPDATE_CNT = "update notice set
visited=visited+1 where no=?";
        final static String NOTICE_DELETE = "delete from notice where no=?";

```

상품 DAO

```

package shop.model;
import shop.dto.Category;
import shop.dto.Fileud;
import shop.dto.Product;
import shop.dto.Receive;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
public class ProductDAO {
        static Connection conn = null;
        static PreparedStatement pstmt = null;
        static ResultSet rs = null;

```



```

    public List<Product> getProductList() {
        List<Product> proList = new ArrayList<>();
        DBConnect con = new MariaDBCon();
conn = con.connect();
        SimpleDateFormat ymd = new SimpleDateFormat("yy-MM-dd");
        try{
pstmt = conn.prepareStatement(DBConnect.PRODUCT_SELECT_ALL);
rs = pstmt.executeQuery();
            while(rs.next()) {
                Product pro = new Product();
pro.setNo(rs.getInt("no"));
pro.setCate(rs.getString("cate"));
pro.setPname(rs.getString("pname"));
pro.setPcomment(rs.getString("pcomment"));
pro.setPlist(rs.getString("plist"));
pro.setPrice(rs.getInt("price"));
pro.setImgSrc1(rs.getString("imgsrc1"));
                Date d = ymd.parse(rs.getString("resdate")); //날짜데이터로 변
경
                String date = ymd.format(d);
pro.setResdate(date);
proList.add(pro);
            }
        } catch (SQLException e) {
System.out.println("sql 에러");
        } catch (ParseException e) {
            throw new RuntimeException(e);
        } finally{
conn.close(rs, pstmt, conn);
        }

        return proList;
    }

    public int updateProduct(Product pro){
        int cnt = 0;
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{
pstmt = conn.prepareStatement(DBConnect.PRODUCT_UPDATE);
pstmt.setString(1, pro.getPname());
pstmt.setString(2, pro.getPcomment());
pstmt.setString(3, pro.getPlist());
pstmt.setInt(4, pro.getPrice());
pstmt.setString(5, pro.getImgSrc1());
pstmt.setString(6, pro.getImgSrc2());
pstmt.setString(7, pro.getImgSrc3());
pstmt.setString(8, pro.getCate());
pstmt.setInt(9, pro.getNo());
cnt = pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }

```

```

} finally{
con.close(pstmt, conn);
}

    return cnt;
}

    public List<Product> getCateProductList(String cate) {
        List<Product> proList = new ArrayList<>();
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        SimpleDateFormat ymd = new SimpleDateFormat("yy-MM-dd");
        try{
            pstmt = conn.prepareStatement(DBConnect.PRODUCT_SELECT_CATE);
            pstmt.setString(1, cate);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                Product pro = new Product();
                pro.setNo(rs.getInt("no"));
                pro.setCate(rs.getString("cate"));
                pro.setPname(rs.getString("pname"));
                pro.setPcomment(rs.getString("pcomment"));
                pro.setPlist(rs.getString("plist"));
                pro.setPrice(rs.getInt("price"));
                pro.setImgSrc1(rs.getString("imgsrc1"));
                Date d = ymd.parse(rs.getString("resdate")); //날짜데이터로 변경
                String date = ymd.format(d);
                pro.setResdate(date);
            }
            proList.add(pro);
        } catch (SQLException e) {
            System.out.println("sql 에러");
        } catch (ParseException e) {
            throw new RuntimeException(e);
        } finally{
            con.close(rs, pstmt, conn);
        }

        return proList;
    }

    public Product getProduct(int no) {
        Product pro = new Product();
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        SimpleDateFormat ymd = new SimpleDateFormat("yy-MM-dd");
        try{
            pstmt = conn.prepareStatement(DBConnect.PRODUCT_SELECT_ONE);
            pstmt.setInt(1, no);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                pro.setNo(rs.getInt("no"));
                pro.setCate(rs.getString("cate"));
            }
        }
    }

```

```

pro.setCateno(rs.getString("cateno"));
pro.setPname(rs.getString("pname"));
pro.setPcomment(rs.getString("pcomment"));
pro.setPlist(rs.getString("plist"));
pro.setPrice(rs.getInt("price"));
pro.setImgSrc1(rs.getString("imgsrc1"));
pro.setImgSrc2(rs.getString("imgsrc2"));
pro.setImgSrc3(rs.getString("imgsrc3"));
        Dated = ymd.parse(rs.getString("resdate")); //날짜데이터로 변
경
        Stringdate = ymd.format(d);
pro.setResdate(date);
}
} catch(SQLException) {
    throw new RuntimeException(e);
} catch(ParseException) {
    throw new RuntimeException(e);
} finally{
con.close(rs, pstmt, conn);
}

    returnpro;
}

    public intaddProduct(Productpro){
        intcnt = 0;
        DBConnectcon = newMariaDBCon();
        Stringsql = DBConnect.PRODUCT_INSERT;
        try{
conn = con.connect();
pstmt = conn.prepareStatement(sql);
pstmt.setString(1, pro.getCate());
pstmt.setString(2, pro.getPname());
pstmt.setString(3, pro.getPcomment());
pstmt.setString(4, pro.getPlist());
pstmt.setInt(5, pro.getPrice());
pstmt.setString(6, pro.getImgSrc1()); // 이미지
pstmt.setString(7, pro.getImgSrc2()); // 소개영상
cnt = pstmt.executeUpdate();
} catch(SQLException) {
    throw new RuntimeException(e);
} catch(Exception) {
    throw new RuntimeException(e);
} finally{
con.close(pstmt, conn);
}

    returncnt;
}

    public intdelProduct(intno){
        intcnt = 0;
        DBConnectcon = newMariaDBCon();
        Stringsql = DBConnect.PRODUCT_DELETE;

```

```

        try{
conn = con.connect();
pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, no);
cnt = pstmt.executeUpdate();
} catch(SQLException) {
        throw new RuntimeException(e);
} catch(Exception) {
        throw new RuntimeException(e);
} finally{
con.close(pstmt, conn);
}

        return cnt;
}

    public List<Category> getCategoryList(){
        List<Category> cateList = new ArrayList<Category>();
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{
pstmt = conn.prepareStatement(DBConnect.CATEGORY_LOAD);
rs = pstmt.executeQuery();
        while(rs.next()){
            Category cate = new Category();
cate.setCno(rs.getString("cno"));
cate.setCname(rs.getString("cname"));
cateList.add(cate);
        }
    } catch(SQLException) {
        throw new RuntimeException(e);
    } finally{
con.close(pstmt, conn);
}

        return cateList;
}

    public int getAmount(int pno){
        int amount = 0;
        DBConnect con = new MariaDBCon();
        try{
conn = con.connect();
pstmt = conn.prepareStatement(DBConnect.INVENTORY_SELECT_ONE);
pstmt.setInt(1, pno);
rs = pstmt.executeQuery();
        if(rs.next()){
amount = rs.getInt("amount");
        }
    } catch(SQLException) {
        throw new RuntimeException(e);
    } finally{
con.close(rs, pstmt, conn);
}
}

```

```

        return amount;
    }

    public int addReceive(Receiver rec) {
        int cnt = 0;
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        try {
            pstmt = conn.prepareStatement(DBConnect.RECEIVE_INSERT);
            pstmt.setInt(1, rec.getPno());
            pstmt.setInt(2, rec.getAmount());
            pstmt.setInt(3, rec.getRprice());
            cnt = pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            con.close(pstmt, conn);
        }

        return cnt;
    }

    public int updateReceive(Receiver rec) {
        int cnt = 0;
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        try {
            pstmt = conn.prepareStatement(DBConnect.RECEIVE_AMOUNT_UPDATE);
            pstmt.setInt(1, rec.getAmount());
            pstmt.setInt(2, rec.getPno());
            cnt = pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } finally {
            con.close(pstmt, conn);
        }

        return cnt;
    }
}

```

상품 SQL

```

final static String PRODUCT_SELECT_ALL = "SELECT * FROM product ORDER BY NO";
// 상품 조회
final static String PRODUCT_SELECT_CATE = "SELECT * FROM product where cate=? ORDER BY NO";
final static String PRODUCT_SELECT_ONE = "SELECT * FROM product where no=?"; // 상품 상세 조회
final static String PRODUCT_SELECT_RECENT = "SELECT * FROM product ORDER BY NO DESC LIMIT 5"; // 최근 상품 조회
final static String PRODUCT_SELECT_BEST = "SELECT * from product where pno IN (SELECT pno FROM payment GROUP BY pno ORDER BY SUM(amount) DESC LIMIT 5)";
// 제일 잘나가는 상품 조회

```

```

        final static String PRODUCT_UPDATE = "update product set pname=?,
pcomment=?, plist=?, price=?, cate=? where no=?";
        final static String PRODUCT_DELETE = "delete from product where no=?";
        final static String PRODUCT_INSERT = "INSERT INTO product VALUES(DEFAULT,
?, '', ?, ?, ?, ?, ?, ' ', DEFAULT)";

```

결제 DAO

```

package shop.model;
import shop.dto.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
public class PaymentDAO {
    static Connection conn = null;
    static PreparedStatement pstmt = null;
    static ResultSet rs = null;
    String sql = "";
    //결제 처리(PaymentDAO.addPayment(pay))
    public int addPayment(Payment pay){
        int cnt = 0;
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{
pstmt = conn.prepareStatement(DBConnect.SERVE_PAYMENT);
pstmt.setString(1, pay.getCid());
pstmt.setInt(2, pay.getPno());
pstmt.setInt(3, pay.getAmount());
pstmt.setString(4, pay.getPmethod());
pstmt.setString(5, pay.getPcom());
pstmt.setString(6, pay.getCnum());
pstmt.setInt(7, pay.getPayprice());
cnt = pstmt.executeUpdate();
        } catch (SQLException) {
            throw new RuntimeException(e);
        } finally{
con.close(pstmt, conn);
        }

        return cnt;
    }
    //출고 처리(PaymentDAO.addServe(serv))
    public int addServe(Serve serv){
        int cnt = 0;
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{

```

```

pstmt = conn.prepareStatement(DBConnect.SERVE_INSERT);
pstmt.setInt(1, serv.getPno());
pstmt.setInt(2, serv.getAmount());
pstmt.setInt(3, serv.getSprice());
cnt = pstmt.executeUpdate();
} catch(SQLException e) {
    throw new RuntimeException(e);
} finally{
    con.close(pstmt, conn);
}

    return cnt;
}

    public int getSno(){
        int sno = 0;
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        try{
            pstmt = conn.prepareStatement(DBConnect.GET_SNO);
            rs = pstmt.executeQuery();
            if(rs.next()){
                sno = rs.getInt("sno");
            }
        } catch(SQLException e) {
            throw new RuntimeException(e);
        } finally{
            con.close(rs, pstmt, conn);
        }

        return sno;
    }

    public int returnPayment(int sno, int pno, int amount, String cid) {
        int cnt = 0;
        DBConnect con = new MariaDBCon();
        conn = con.connect();
        try{
            conn.setAutoCommit(false);
            //1. 반품시 결제 내용 제거
            pstmt = conn.prepareStatement(DBConnect.RETURN_PAYMENT);
            pstmt.setInt(1, sno);
            cnt = pstmt.executeUpdate();
            //2. 반품시 배송 정보 삭제
            pstmt = conn.prepareStatement(DBConnect.RETURN_DELEVERY);
            pstmt.setInt(1, sno);
            cnt = cnt + pstmt.executeUpdate();
            //3. 반품시 출고 삭제
            pstmt = conn.prepareStatement(DBConnect.RETURN_SERVE);
            pstmt.setInt(1, sno);
            cnt = cnt + pstmt.executeUpdate();
            //4. 반품시 장바구니에 다시 담기
            pstmt = conn.prepareStatement(DBConnect.RETURN_CART);
            pstmt.setString(1, cid);

```

```

pstmt.setInt(2, pno);
pstmt.setInt(3, amount);
cnt = cnt + pstmt.executeUpdate();
conn.commit();
conn.setAutoCommit(true);
} catch(SQLException e) {
    try{
conn.rollback();
} catch(SQLException ex) {
    throw new RuntimeException(ex);
}
    throw new RuntimeException(e);
} finally{
con.close(pstmt, conn);
}
    return cnt;
}

    public Payment getPayment(int sno){
        Payment pay = new Payment();
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{
pstmt = conn.prepareStatement(DBConnect.PAYMENT_SELECT_ONE);
pstmt.setInt(1, sno);
rs = pstmt.executeQuery();
            if(rs.next()){
pay.setSno(rs.getInt("sno"));
pay.setCid(rs.getString("cid"));
pay.setPno(rs.getInt("pno"));
pay.setAmount(rs.getInt("amount"));
}
        } catch(SQLException e) {
            throw new RuntimeException(e);
        } finally{
con.close(rs, pstmt, conn);
}
        return pay;
    }

    public int getPaymentBuyCustom(int pno, String cid) {
        int check = 0;
        DBConnect con = new MariaDBCon();
conn = con.connect();
        try{
pstmt = conn.prepareStatement(DBConnect.PAYMENT_SELECT_CID_PNO);
pstmt.setString(1, cid);
pstmt.setInt(2, pno);
rs = pstmt.executeQuery();
            if(rs.next()) {
check = 1;
System.out.println(cid + "는 " + pno + " 상품을 구매한 유저 입니다.");

```



```

    }
    } catch(SQLException e) {
        throw new RuntimeException(e);
    }

    return check;
}

    public List<PaymentVO> getCidPaymentList(String cid){
        List<PaymentVO> payList = new ArrayList<>();
        DBConnect con = new MariaDBCon();
    conn = con.connect();
        try{
    pstmt = conn.prepareStatement(DBConnect.PAYMENT_SELECT_CID);
    pstmt.setString(1, cid);
    rs = pstmt.executeQuery();
        while(rs.next()){
            PaymentVO pay = new PaymentVO();
    pay.setSno(rs.getInt("sno"));
    pay.setCid(rs.getString("cid"));
    pay.setPno(rs.getInt("pno"));
    pay.setAmount(rs.getInt("amount"));
    pay.setPmethod(rs.getString("pmethod"));
    pay.setPcom(rs.getString("pcom"));
    pay.setCnum(rs.getString("cnum"));
    pay.setPayprice(rs.getInt("payprice"));
    pay.setPname(getVOPname(pay.getPno()));
    pay.setPstate(getVOState(pay.getSno()));
    payList.add(pay);
        }
    } catch(SQLException e) {
        throw new RuntimeException(e);
    }
    } finally{
    con.close(rs, pstmt, conn);
    }

        return payList;
    }

    public String getVOPname(int pno){
        ProductDAO dao = new ProductDAO();
        Product pro = dao.getProduct(pno);
        return pro.getPname();
    }

    public int getVOState(int sno){
        DeliveryDAO dao = new DeliveryDAO();
        Delivery del = dao.getBySnoDelivery(sno);
        return del.getPstate();
    }

    public List<Sales> getSales() {
        List<Sales> salesList = new ArrayList<>();
        DBConnect con = new MariaDBCon();

```

```

conn = con.connect();
    try{
pstmt = conn.prepareStatement(DBConnect.SALES_SELECT_ALL);
rs = pstmt.executeQuery();
        while(rs.next()) {
            Sales s = new Sales();
s.setPno(rs.getInt("pno"));
s.setCnt(rs.getInt("total_amount"));
s.setMoney(rs.getInt("total_cost"));
s.setPname(rs.getString("pname"));
salesList.add(s);
        }
    } catch(SQLException e) {
        throw new RuntimeException(e);
    }

    returnsalesList;
}
}

```

결제 SQL

//입고처리패턴

```

    final static String PRODUCT_RECEIVE = "insert into receive values
(default, ?, ?, ?, default)";

```

```

    final static String CATEGORY_LOAD = "SELECT * FROM CATEGORY";

```

//출고 처리 패턴

```

    final static String SERVE_PAYMENT = "insert into payment values (default,
?, ?, ?, ?, ?, ?, ?, ' ');

```

```

    final static String SERVE_INSERT = "insert into serve values (default, ?,
?, ?, default)";

```

```

    final static String DELIVERY_INSERT = "insert into delivery values
(default, ?, ?, ?, ?, ' ', ' ', default, default, ' ', ' ');

```

```

    final static String CART_DELETE = "delete from cart where cartno=?";

```

```

    final static String GET_SNO = "select sno from payment order by sno desc
limit 1";

```

//반품 처리 패턴

```

    final static String RETURN_PAYMENT = "delete from payment where sno=?";

```

```

    final static String RETURN_RECEIVE = "insert into receive values (default,
?, ?, ?, default)";

```

```

    final static String RETURN_SERVE = "delete from serve where sno=?";

```

```

    final static String RETURN_CART = "insert into cart values (default, ?, ?,
?)";

```

```

    final static String RETURN_DELEVERY = "delete from delivery where sno=?";

```

//배송 등록 처리

//배송 등록 처리

```

    final static String DELIVERY_PRO = "update delivery set pcom=?, ptel=?,
pstate=?, rdate=?, bcode=? where dno=?";

```

// 배송 조회

```

    final static String DELIVERY_SELECT_ALL = "select * from delivery where
pstate=?";

```

```
final static String DELIVERY_SELECT_ONE = "select * from delivery where  
dno=?";  
final static String DELIVERY_SELECT_BUY = "select * from delivery_info  
where pno=? and cid=? and pstate=3";  
final static String DELIVERY_SELECT_SNO = "select * from delivery where  
sno=?";  
final static String DELIVERY_BUY = "update delivery set pstate=3 where  
sno=?";  
// 배송 뷰 조회  
final static String DELIVERY_VIEW = "select * from delivery_info where  
sno=?";  
// 재고 조회  
final static String INVENTORY_SELECT_ALL = "select * from inventory order  
by pno desc";  
final static String INVENTORY_SELECT_ONE = "select * from inventory where  
pno=?";  
// 입고 처리  
final static String RECEIVE_INSERT = "insert into receive values (default,  
?, ?, ?, default)";  
final static String RECEIVE_AMOUNT_UPDATE = "update receive set  
amount=amount+? where pno=?";  
// 결제 정보  
final static String PAYMENT_SELECT_LIST = "select * from payment order by  
sno desc";  
final static String PAYMENT_SELECT_ONE = "select * from payment where  
sno=?";  
final static String PAYMENT_SELECT_CID = "select * from payment where  
cid=?";  
final static String PAYMENT_SELECT_CID_PNO = "select * from payment where  
cid=? and pno=?";
```